60469690

CONTROL DATA CORPORATION

CDC® CYBER 170 COMPUTER SYSTEM MODEL 835

CDC® CYBER 180 COMPUTER SYSTEM MODEL 835

VIRTUAL STATE

VOLUME I
SYSTEM DESCRIPTION
FUNCTIONAL DESCRIPTIONS

HARDWARE REFERENCE MANUAL

REVISION RECORD

REVISION	DESCRIPTION
A (04-84)	Manual released.

Publication No.
60469690

REVISION LETTERS I, O, Q, S, X AND Z ARE NOT USED.

Address comments concerning this manual to:
Control Data Canada, Ltd.
Toronto Publications (210)
1855 Minnesota Court
Mississauga, Ontario
Canada L5N 1K7

or use Comment Sheet in the back of this manual.

LIST OF EFFECTIVE PAGES

New features, as well as changes, deletions, and additions to information in this manual, are indicated by bars in the margins or by a dot near the page number if the entire page is affected. A bar by the page number indicates pagination rather than content has changed.

PAGE	REV
Front Cover	A
Title Page	A
2	A
3/4	A
5	A
6	A
7	A
8	A
1-1-1	A
1-1-2	A
1-1-3	A
1-1-4	A
1-1-5	A
1-1-6	A
1-1-7	A
1-1-8	A
1-1-9	A
1-1-10	A
1-1-11	A
I-2-1	A
I-2-2	A
I-2-3	A
I-2-4	A
I-2-5	A
I-2-6	A
I-2-7	A
I-2-8	A
I-2-9	A
I-2-10	A
I-2-11	A
I-2-12	A
I-2-13	A
I-2-14	A
I-2-15	A
I-2-16	A
I-2-17	A
I-2-18	A
I-2-19	A
I-2-20	A

PAGE	REV
I-2-21	A
I-2-22	A
I-2-23	A
I-2-24	A
I-2-25	A
I-2-26	A
I-2-27	A
I-2-28	A
I-2-29	A
I-2-30	A
I-2-31	A
I-2-32	A
I-2-33	A
I-2-34	A
I-2-35	A
Comment Sheet	A

PAGE	REV

PAGE	REV

PAGE	REV

PREFACE

This manual contains hardware reference information for the CDC CYBER 170/180 Model 835 Computer System in its virtual state of operation.

This manual provides model-dependent information regarding the system description and functional descriptions of the computer system hardware. Additional system hardware information is available in manuals listed in the System Publication Index on the following page.

AUDIENCE

This manual is for use by customers as well as marketing and training personnel who want a general, yet technical, description of the computer system.

Other manuals applicable to the model 835 computer system are:

Control Data Publication	Publication Number
CYBER 170/180 Computer Systems Models 835, 845, and 855 Power and Warning System	60455920
CYBER 170/180 Computer System Models 835, 845, and 855 Cooling System	60455930
CYBER 170/180 Computer System Models 835, 845, and 855 Hardware Operator's Guide	60458390
NOS/VE Operations Usage	60463914

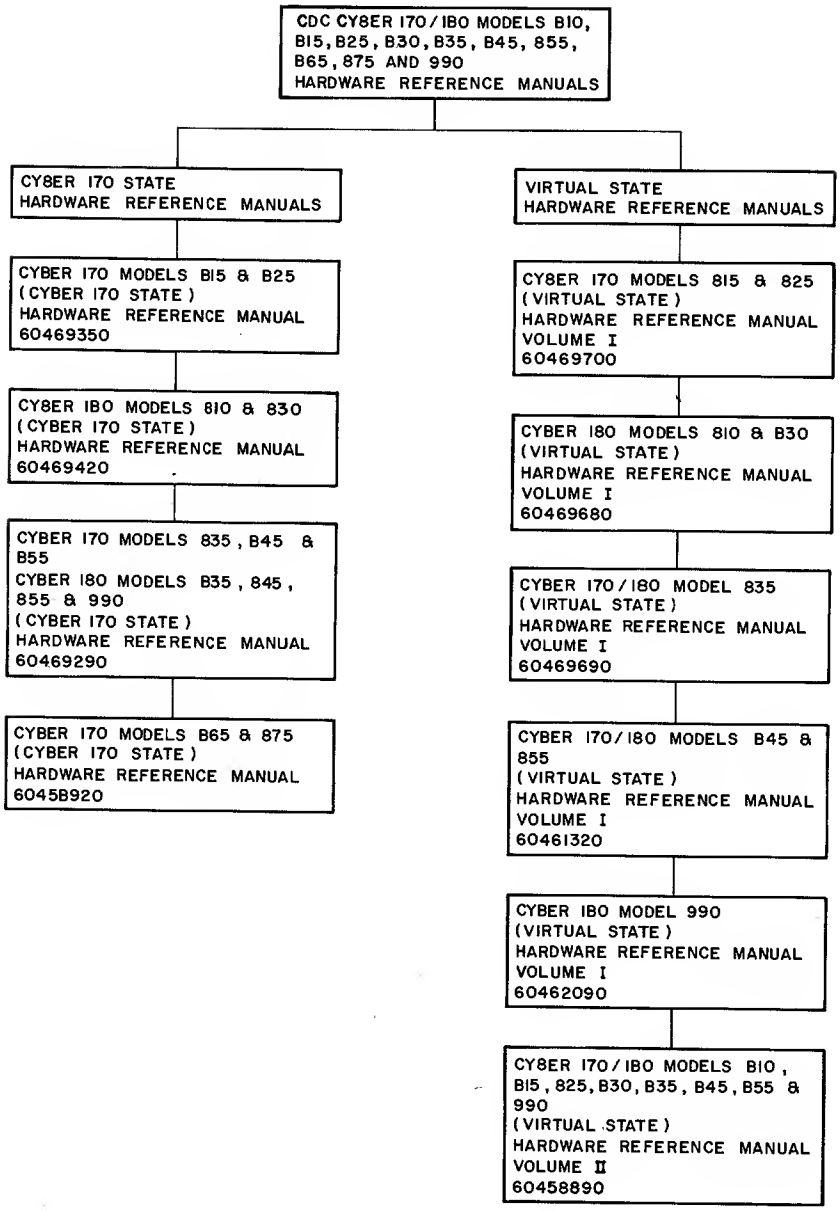
Publication ordering information and latest revision levels are available from the Literature Distribution and Services catalog, publication number 90310500.

This manual does not contain an index. An index will be available in the next revision.

WARNING

This equipment generates, uses, and can radiate radio frequency energy, and if not installed and used in accordance with the instructions manual, may cause interference to radio communications. It has been tested and found to comply with the limits for a Class A computing device persuant to Subpart J of Part 15 of the FCC Rules which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference in which case the user, at his own expense, will be required to take whatever measures may be required to correct the interference.

SYSTEM PUBLICATION INDEX



C1043

CONTENTS

FIGURES

TABLES

SYSTEM DESCRIPTION 1

This section introduces the CYBER 170/180 Computer System Model 835 by identifying its physical and functional characteristics and by providing descriptions of the major system components.

INTRODUCTION

Model 835, shown in figure I-1-1, is a medium-scale, high-speed computer system which can be used for both business and scientific applications. The system includes the following components:

- Central processor (CP)
- Central memory (CM)
- Input/output unit (IOU)

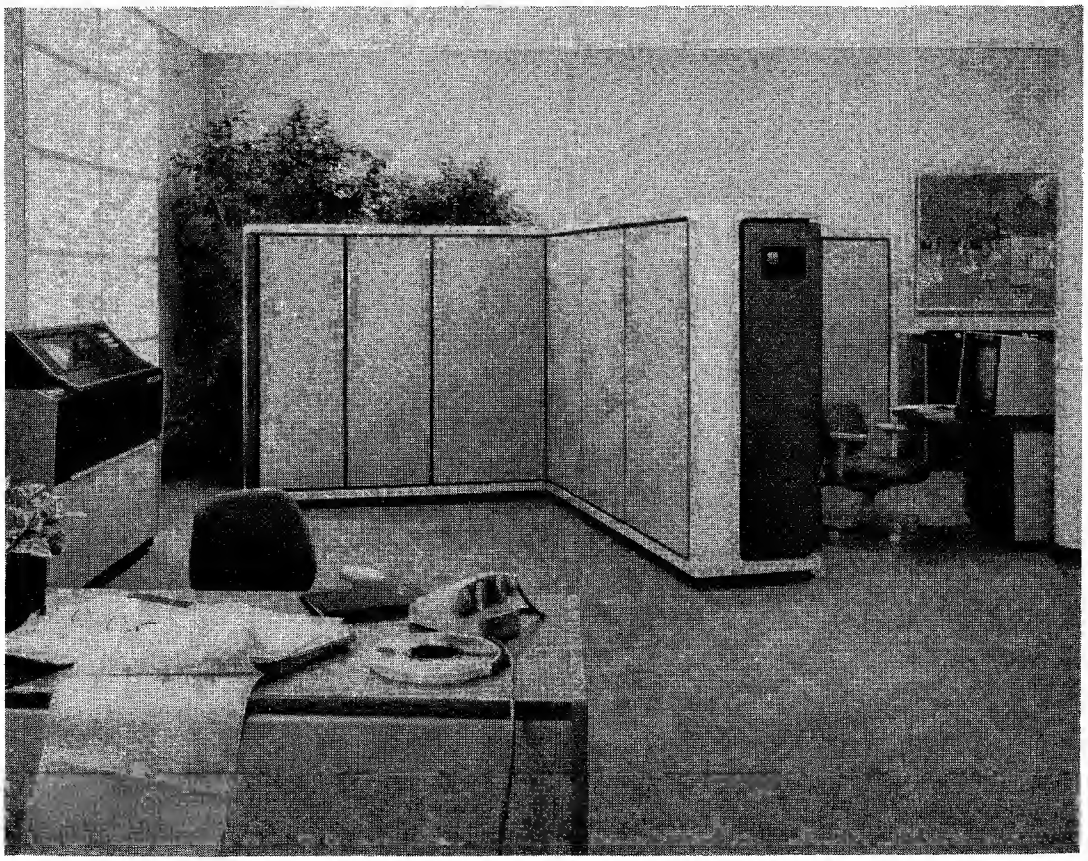


Figure I-1-1. Model 835 Computer System

PHYSICAL CHARACTERISTICS

The mainframe configuration for model 835, shown in figure I-1-2, includes an interconnected three-section cabinet for the CP, CM, and IOU. A display station is also required for system operation. Each cabinet section contains a logic chassis with built-in circuit boards. The logic chassis in the IOU also contains a deadstart panel with initialization and maintenance controls and displays.

Each cabinet section also contains a cooling unit to cool the logic chassis, an ac/dc control section with voltage margin testing facilities, and dc power supplies. For additional power or cooling information, refer to the cooling system and power system manuals listed in the preface.

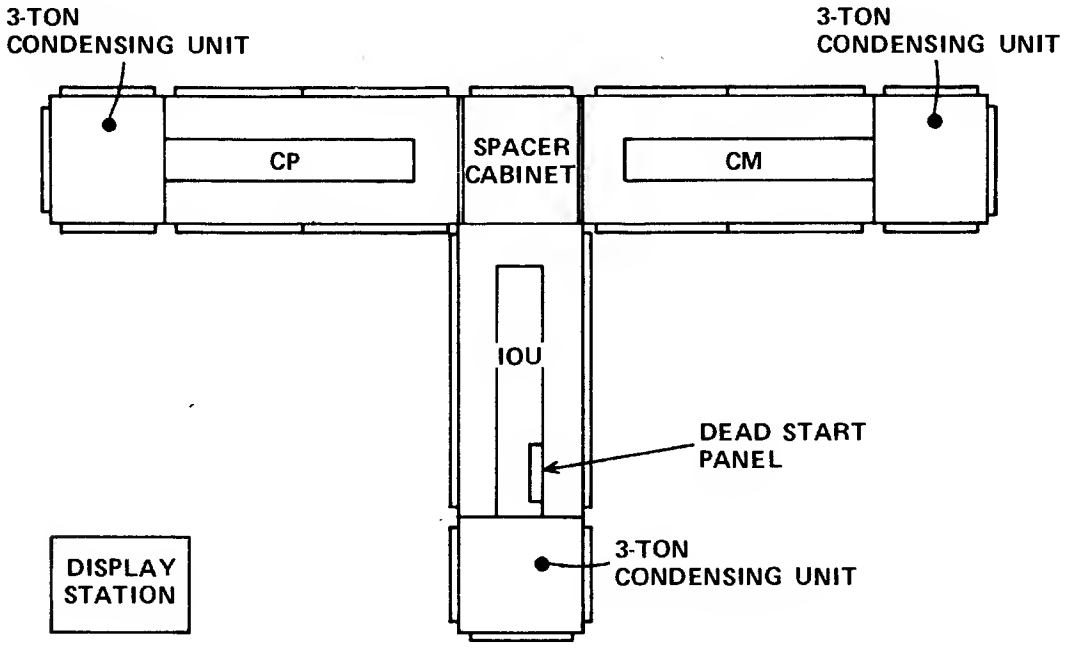


Figure I-1-2. Chassis Configuration (Top Cutaway View)

FUNCTIONAL CHARACTERISTICS

To achieve high computation speeds, the computer system uses emitter-coupled logic (ECL). High speed is also the objective of the CP design. The CP design is based on the assumption that both instructions and data are, in most cases, accessed from successive memory locations. Accordingly, the CP prefetches both instructions and data expected to be used next while the current instruction is being processed.

The CP supports two states of operation.

- Virtual State — Operation as a computer with virtual memory byte addressing using the Virtual State instruction set and data formats. Virtual State is the native operating state of the CP.
- CYBER 170 State — Operation as a computer with real memory word addressing using the CYBER 170 State CP instruction set and data formats.

The Virtual State and CYBER 170 State environments may be present at the same time with the CP executing in either environment. NOS/VE is the operating system of Virtual State; NOS is the operating system of CYBER 170 State.

The semiconductor memory is divided into eight independent banks to minimize conflicts among central memory requests. These banks may all be simultaneously in the process of completing read and write requests. System input/output speeds are determined by the capabilities of existing external devices.

CP GENERAL FUNCTIONAL CHARACTERISTICS

The following CP characteristics are common to both Virtual State and CYBER 170 State:

- 56-nanosecond clock period
- Microprogram control
- Cache buffer memory, with data lookahead
- Instruction and branch target instruction lookahead
- Separate arithmetic units for fixed-/floating-point binary and decimal data processing
- Hardware data format checking, conversion, and editing
- Parity checking on selected data and address paths

CP VIRTUAL STATE FUNCTIONAL CHARACTERISTICS

The following CP characteristics are exclusive to Virtual State.

Main Registers

The following CP registers hold most of the operands and addresses used for computational purposes:

- Sixteen 48-bit address (A) registers
- 64-bit program address (P) register
- Sixteen 64-bit operand (X) registers

Processing Capabilities

The CP processing capabilities have the following characteristics:

- 64-bit internal word (8 bytes)
- Packed instructions (16- and 32-bit instructions in 64-bit words)
- Integer arithmetic (32/64-bit operands)
- Floating-point arithmetic (12-bit exponent plus sign bit, 48/96-bit coefficient plus sign bit)
- Business data processing (11 decimal data types and an alphanumeric data type supported by move/compare/edit instructions)
- Call and return mechanism
- Load and store fields of one through eight bytes
- Extract and insert strings of 1 through 64 bits
- Load and store multiple address (A) and operand (X) registers
- Process immediate data from instruction word

Modes of Operation and Interrupts

The CP modes of operation and CP interrupt structure have the following characteristics:

- Monitor and job modes of operation
- Exchange instructions causing exchange of operating mode and executing process
- Trap interrupts on monitor mode conditions, with trap handled within present operating mode
- Trap interrupts on job mode conditions, with trap handled within present operating mode
- Exchange interrupts on job mode conditions, with an exchange to monitor mode for interrupt processing

Program Monitoring

CP program monitoring has five maskable classes of debug interrupts on up to 32 process virtual address ranges.

Access Protection

The CP security features include the following:

- Controlled access to segments which a process may access
- Fifteen rings of protection
- Segment locks and keys
- Eight types of segment access (read/write/execute with subdivisions)
- Controlled and protected entry points into shared code

CP CYBER 170 STATE FUNCTIONAL CHARACTERISTICS

For CP characteristics exclusive to CYBER 170 State, refer to the appropriate CYBER 170 State hardware reference manual listed in the System Publication Index.

CM FUNCTIONAL CHARACTERISTICS

The CM has the following functional characteristics:

- 72-bit data word [64 data bits and 8 single-error correction double-error detection (SECDED) bits]
- 524K words (4 Megabytes) of refresh-type semiconductor memory, options available to 2097K words (16 Megabytes)
- Organization of eight independent banks
- Directly addressable process virtual address space of up to 4096 segments, with up to 4 billion bytes per segment
- System virtual address space of up to 65K segments
- Real memory page size ranging from 2K to 16K bytes
- Bounds register to limit write access
- 56-nanosecond clock period
- Maximum data transfer rate of one word every 56 nanoseconds
- 672-nanosecond read access time
- 448-nanosecond read/write cycle time
- 896-nanosecond partial-write cycle time
- Read and write data queuing capability
- SECDED CM data verification
- Parity checking on all major data, address, and control paths

IOU FUNCTIONAL CHARACTERISTICS

The IOU has the following functional characteristics:

- Ten peripheral processors (PPs), 15-PP and 20-PP options available. Each PP has a 4K independent memory (PPM) composed of 16-bit words. The PPs are compatible with CYBER 170 State PP instructions, data formats, and channels.
- Execution of 12- or 16-bit PP code.
- Port to central memory.
- Twelve I/O channels available to external devices. 24-channel option available.
- External interface to real-time clock, display controller, and two-port multiplexer.
- Bounds register controlling write access to CM.
- Parity checking on all major data and address paths.
- Maintenance channels giving PPs access to CP, CM, and IOU registers to perform system initialization and maintenance functions.
- Operating speed of 250 nanoseconds and a minor cycle of 50 nanoseconds.

MAJOR SYSTEM COMPONENT DESCRIPTIONS

The major system components are:

- Central processor (CP)
- Central memory (CM)
- Input/output unit (IOU)

The rest of this section provides brief descriptions of the major system components. The descriptions relate to the computer system block diagram shown in figure I-1-3.

CENTRAL PROCESSOR

The CP consists of the following:

- Instruction section
- Operand issue section
- Execution section
- Segment map
- Local memory
- Addressing section
- Business data processing (BDP) section
- Maintenance access control (MAC)

The CP is isolated from the input/output unit and is thus able to carry on computation or character manipulation unencumbered by I/O requirements.

Instruction Section

The instruction section directs the arithmetic and manipulative functions for instruction execution. The instruction section does the following:

- Initializes registers, controls, and memories
- Stores, accesses, decodes, and initiates a microprogram which controls CP operation in both CYBER 170 State and Virtual State
- Prefetches and disassembles instructions from CM
- Initiates interrupts when an error or exception condition occurs during instruction execution

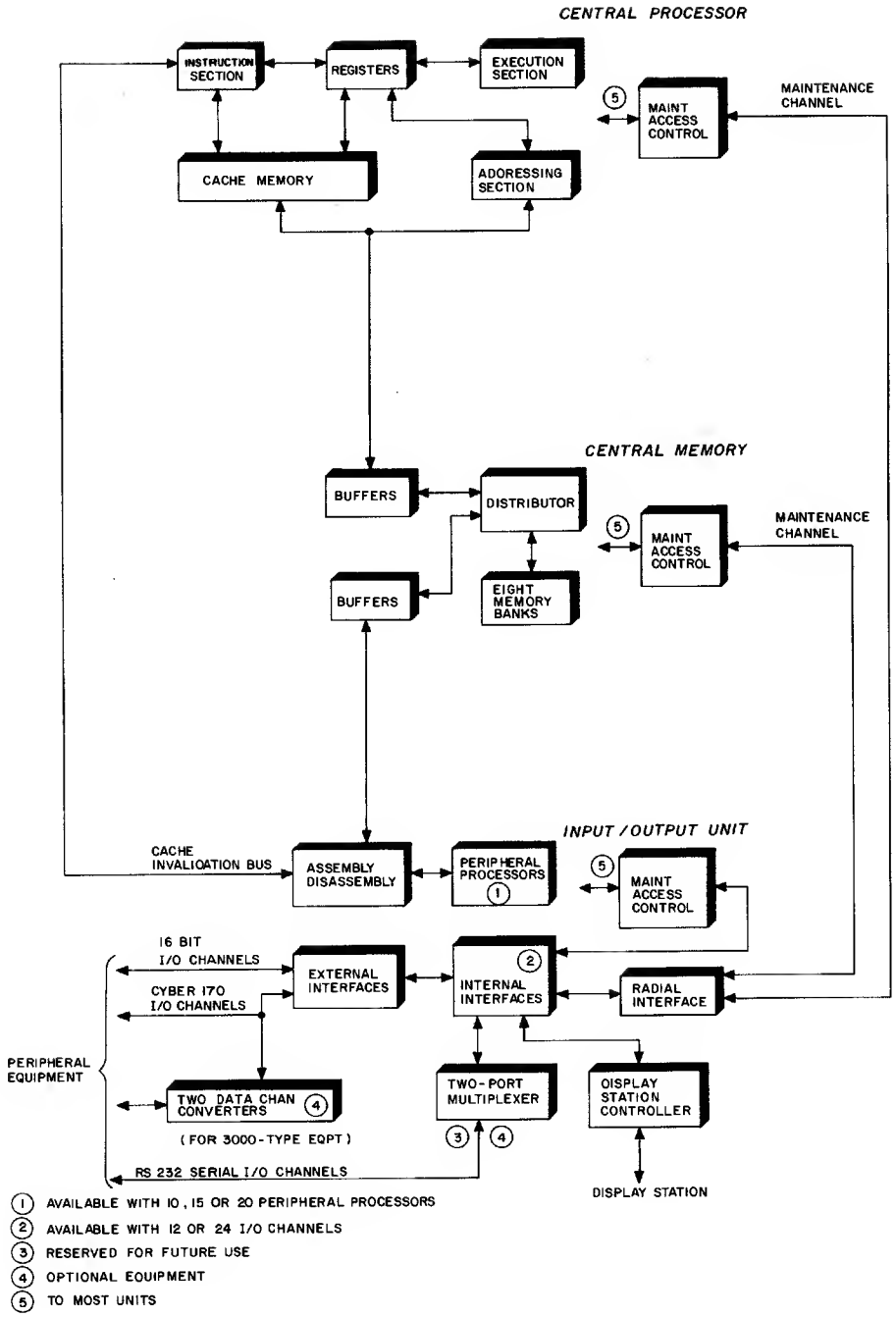


Figure I-1-3. Model 835 Computer System Block Diagram

Operand Issue Section

The operand issue section contains the registers of the two CP register categories:

- Process state registers
- Processor state registers

These registers are located in the 64-word register file or throughout the hardware as various live registers.

The process state registers contain current process (job) information. This includes operands and exchange package data required for instruction execution. These registers change upon the switching of processes. The 64-word register file contains the operating registers for instructions. These are the A, B, and X registers for CYBER 170 State instructions, and the A and X registers for Virtual State instructions. Refer to Process State Registers described under the heading Operand Issue Section in section 2.

The processor state registers contain information related to the system and to the CP hardware. The live registers are processor state registers which supply data for various operations and collect data from the CP operating environment. Refer to Processor State Registers described under the heading Operand Issue Section in section 2.

Execution Section

The execution section consists of the arithmetic and logical network (ALN). The ALN performs the logical, arithmetic, shift, and character manipulation functions for instruction execution and address formation. These functions include data formatting, positioning, testing, and comparison as well as data streaming control for BDP operations.

The ALN performs operations on values supplied by the operand issue section or, in some cases, the BDP section. (The BDP section performs most BDP operations independently.) All results return to the operand issue section except for exception conditions, which are sent to the instruction section to initiate interrupt handling.

The ALN consists of a general network and a multiply/divide network. The general network adds and subtracts integers and FP coefficients, and performs exponent arithmetic associated with all FP operations. It also performs Boolean, FP normalize, shift, and conditional branch test functions. The multiply/divide network forms Virtual State integer products and FP product coefficients for both CYBER 170 State and Virtual State.

Segment Map

Segment map contains the hardware to convert process virtual addresses (PVAs) to system virtual addresses (SVAs). Segment map performs this by translating the user's segment number to an active segment identifier (ASID).

To reduce CM access time, segment map contains up to 32 of the most recently used 64-bit segment descriptor table (SDT) entries from the SDT in central memory. If the requested entry is not available, segment map obtains it from the SDT.

Segment map also performs all of the access validations for addressing memory. The access validations include security ring tests, key and lock tests, and read/write/execute privilege validity tests.

Local Memory

Local memory contains the hardware to translate SVAs received from segment map to real memory addresses (RMAs). To reduce CM access time, local memory also contains two buffer memories: cache memory and page map memory. Cache memory contains 2048 or 4096 words of the most recently used entries in system virtual memory. Page map contains up to 128 of the most recently used page descriptors from the system page table.

For a data request, the CP simultaneously tests cache memory and page map to see if the requested SVA is present. If a cache memory hit occurs, the CP reads the requested data from cache memory. If the data is not in cache, the system page table (SPT) is required to convert an SVA to an RMA to access CM. The presence of the most recently used SPT entries in page map accelerates addressing of CM. The page map test, now relevant because the cache test was unsuccessful, searches the SPT for the requested page descriptor. If the test is unsuccessful, the CP retrieves the page with the requested data from CM.

Addressing Section

The addressing section performs the following functions:

- Forms the byte number of addresses sent to local memory for register file data or BDP stream data
- Provides assembly and disassembly data paths and control for data streams from local memory to the BDP section, and vice versa
- Processes the load and store bit, byte, and word instructions

Business Data Processing (BDP) Section

The BDP section performs BDP operations on 12 types of binary, alphanumeric, and decimal data. Although the BDP section performs most BDP operations independently, for some operations it may require processing assistance from the ALN in the execution section.

Maintenance Access Control (MAC)

The MAC performs initialization and maintenance operations in the CP. These operations, controlled by a dedicated peripheral processor in the IOU, include the following:

- Initialize registers, controls, and memories
- Communicate CP error and status information to the IOU when requested by a PP
- Read and write CP-resident registers and memories
- Reconfigure hardware
- Start and stop CP execution

The IOU sends codes to MAC to indicate the operations to be performed and the affected memory device, registers, and addresses.

CENTRAL MEMORY (CM)

The CM, shown in figure I-1-3, is a refresh-type metal-oxide semiconductor (MOS) memory organized into eight independent banks. Memory data words comprise 64 data bits plus 8 SECDED bits. The SECDED bits allow CM to correct single-bit failures and detect double-bit failures during a read operation before sending the data to the requesting unit. For double-bit failures, the CP logs the error and the system software determines corrective action.

INPUT/OUTPUT UNIT (IOU)

The IOU, shown in figure I-1-3, consists of the following:

- Ten logically independent peripheral processors (PPs). Options are available to increase the total to 15 or 20 PPs.
- Internal interface to 12 I/O channels. 24-channel option is available.
- External interfaces to I/O channels.
  - Eight 12-bit channels
  - Display controller interface (radial) (channel 10_8)
  - Maintenance channel interface (radial) (channel 17_8)
  - Real-time clock interface (channel 14_8)
  - Two-port multiplexer interface (channel 15_8)
- Cache invalidation bus to CP.
- Two ports to central memory.
- Bounds register to limit writes to CM.

The PPs are organized in groups of five, called barrels. The PPs in a barrel time-share common hardware. Each PP has its own 4K independent memory and communicates with all I/O channels and with central memory.

DISPLAY STATION

The display station, required for system operation, is a two-way communications facility between the system and the computer operator. The receipt of symbol and position information from the computer enables the display of program information on a 21-inch cathode ray tube (CRT). The station also contains an alphanumeric keyboard which enables an operator to send data to the computer.

The keyboard and CRT combination permits the computer operator to monitor and control system operation. The computer outputs two alternate nonrelated data streams. The display station keyboard has a switch which enables the operator to select either of the data streams or to select both for presentation on the CRT. For further display station information, refer to Display Station Programming in Volume II, section 2, listed in the System Publication Index.

FUNCTIONAL DESCRIPTIONS 2

This section describes the CP operating states, environment interface (EI), and intrastate modes of operation. This section also provides functional descriptions of the central processor (CP), central memory (CM), and input/output unit (IOU). The CP, CM, and IOU functional descriptions relate to the block diagram shown in figure I-1-3 of section 1. Functional descriptions for the cooling system are in the respective manual listed in the System Publication Index in the preface.

CP OPERATING STATES

The CP supports two states of operation: Virtual State and CYBER 170 State. Virtual State uses the Virtual State instruction set and data formats. CYBER 170 State uses the CYBER 170 State CP instruction set and data formats.

As stated earlier, although Virtual State is the native operating state of the system, both environments may be present at the same time with the CP executing in either environment. Portions of Virtual State support and track CYBER 170 State operation. However, Virtual State is transparent to the CYBER 170 State operating system and any user jobs executing in the CYBER 170 State environment.

ENVIRONMENT INTERFACE (EI)

The EI is a Virtual State operating system routine that provides the interface between CYBER 170 State and Virtual State. EI directly supports the CYBER 170 State environment by simulating those portions of that environment not provided by the CP hardware.

The basic EI tasks include:

- Supporting system initialization and deadstart of the CYBER 170 State environment
- Simulating CYBER 170 State hardware and software error processing
- Simulating certain CYBER 170 State instructions unimplemented by the CP hardware
- Processing Virtual State errors occurring in CYBER 170 State

Refer to Intrastate Modes of Operation, following, for additional information on Virtual State EI operations.

INTRASTATE MODES OF OPERATION

Virtual State and CYBER 170 State each have two modes of execution: job mode and monitor mode. Job mode executes programmed sequences of instructions (jobs) in the CP. Monitor mode executes various operating system routines (for example, job sequencer, trap handler, and memory manager) which control the loading, scheduling, executing, and outputting of user jobs. The monitor mode routines are always available to the CP when it requests any type of monitor mode intervention.

The CYBER 170 State environment exists within Virtual State job mode. The operating system supports this environment somewhat like a special purpose Virtual State job. CYBER 170 State has the characteristics of CYBER 170 computer system CPs. For detailed information on CYBER 170 State, refer to the appropriate CYBER 170 State hardware reference manual.

The CP always operates in one of the following environments:

- Virtual State job mode
- Virtual State monitor mode
- CYBER 170 State job mode
- CYBER 170 State monitor mode

These four modes, plus specific EI operations, are briefly described in the following paragraphs. The interaction among operating states, intrastate modes of operation, and EI are shown in figure I-2-1.

VIRTUAL STATE JOB MODE

Virtual State job mode executes programmed sequences of Virtual State instructions in the CP. While in Virtual State job mode, exchange iterrupts or an exchange instruction cause an exchange to Virtual State monitor mode.

CYBER 170 State monitor and job modes exist within Virtual State job mode. Refer to the corresponding headings in this section for further detail.

EI Operations in Virtual State Job Mode

EI operations occur in both Virtual State job and monitor modes. (Monitor mode operations are described under Virtual State Monitor Mode, following.) The primary EI job mode task is to perform CYBER 170 State interrupt processing which comprises:

- Simulating CYBER 170 State error exit conditions
- Trapping from CYBER 170 State to routines in Virtual State job mode that:
  - Execute CYBER 170 State compare/move instructions
  - Purge cache memory of memory word blocks in CM and external extended memory (UEM)
  - Ready the CP for transition from Virtual State to CYBER 170 State
  - Transfer blocks of CM or UEM words either within the CYBER 170 State environment or between operating states

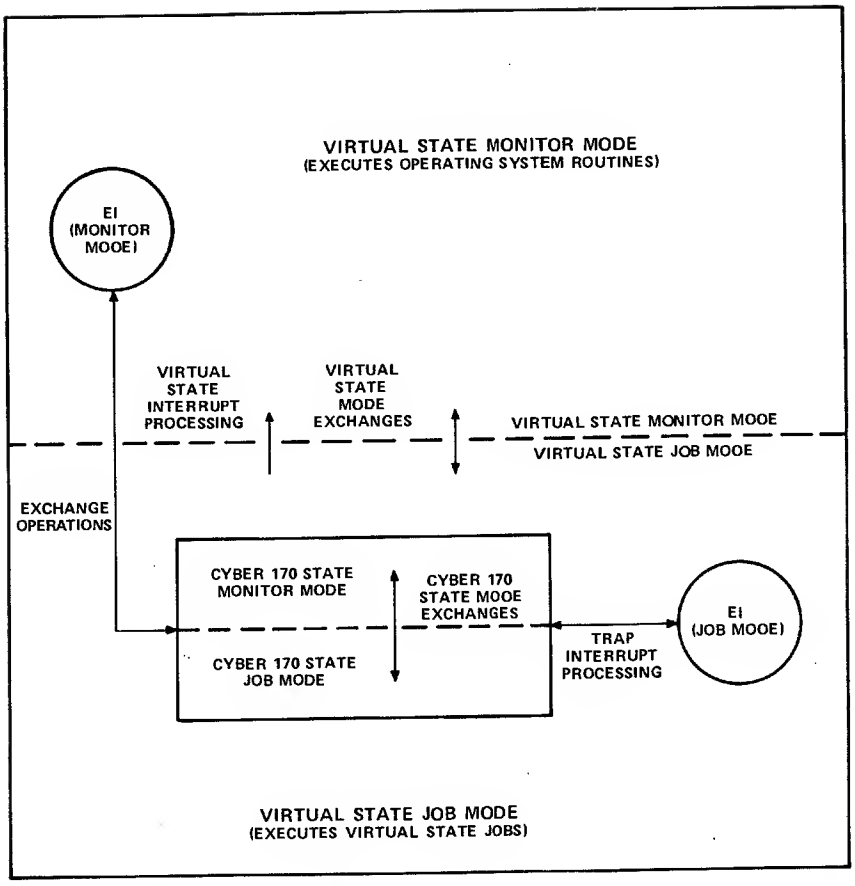


Figure I-2-1. Operating States Interaction

VIRTUAL STATE MONITOR MODE

Virtual State monitor mode executes operating system routines that perform Virtual State monitor activities. This mode performs exchange and trap interrupt processing, simulates certain CYBER 170 State instructions, executes restricted CP instructions, and processes hardware and software errors detected in both Virtual State and CYBER 170 State.

EI Operations in Virtual State Monitor Mode

The primary EI monitor mode task is to perform exchange operations within CYBER 170 State if hardware or software errors are detected in CYBER 170 State job or monitor mode.

CYBER 170 STATE JOB MODE

CYBER 170 State job mode executes programmed sequences of CYBER 170 State instructions in the CP. This mode, along with CYBER 170 State monitor mode, exists within Virtual State job mode.

CYBER 170 STATE MONITOR MODE

CYBER 170 State monitor mode executes operating system routines that perform CYBER 170 State monitor activities.

Mode Switching Within CYBER 170 State

Within CYBER 170 State, switching between monitor mode and job mode does not require an exchange to Virtual State monitor mode. This transition may be caused by:

Job-to-monitor	PP-directed CYBER 170 State exchange request
	CYBER 170 State exchange jump instruction
	Hardware or conditional software error
Monitor-to-job	CYBER 170 State exchange jump instruction

Transferring from CYBER 170 State to Virtual State

The following CP conditions cause a transfer (exchange or trap) to Virtual State from CYBER 170 State monitor or job mode:

- Explicit trap instruction (trap)
- Request for compare/move instruction (trap)
- Conditional software error in CYBER 170 State monitor mode (exchange)
- Illegal instruction in CYBER 170 State monitor mode (exchange)
- Virtual State errors detected in CYBER 170 State monitor or job mode (exchange)

For further information, refer to Exception Handling in CYBER 170 State in volume II.

CENTRAL PROCESSOR (CP)

The CP consists of the instruction section, operand issue section, execution section, segment map, local memory, addressing section, business data processing (BDP) section, and maintenance access control (MAC).

INSTRUCTION SECTION

The instruction section consists of logic for instruction lookahead, decode, and initiation. The instruction section implements the CYBER 170 State and Virtual State instruction sets by initiating microprogram sequences that obtain instruction operands and provide the control signals for execution. It also performs CP interrupt handling by initiating an interrupt-handling routine when an error or exception condition occurs during the execution of an instruction.

Instruction Lookahead

Model 835 instruction lookahead hardware (ILH) prefetches instruction words to make the next instruction immediately available when execution of the previous instruction is complete. This includes target instructions for conditional branch instructions, and the next instruction after a BDP instruction which is followed by data descriptors. To accomplish this, ILH reads instructions from cache/CM into a three-word, first-in-first-out buffer and prescans the buffer for branch and BDP instructions.

When ILH detects a conditional branch, it reads two instruction words from the target address into the branch buffer, and holds these until the branch is resolved. If the branch takes place, the next two executable instruction words are immediately available.

Instruction Decode and Initiation

In both Virtual State and CYBER 170 State, instruction words read from CM contain from two to four instructions combined in a parcel arrangement. The instruction section decodes the CM instruction word into its separate instructions and issues control information to functional units in the execution section to start instruction execution.

In CYBER 170 State, a two-parcel instruction is not permitted to cross a word boundary, while it is permitted in Virtual State. During Virtual State BDP operations, one- or two-parcel BDP descriptors follow the instruction parcels through execution.

Interrupt Handling

The instruction section can initiate an interrupt-handling routine when an error or exception condition occurs while an instruction is executing. The error/exception conditions are a combination of stackable or unstackable conditions, allowing the instruction section to interrupt selectively.

The error and exception conditions accumulate in the monitor and user condition registers. For more information, refer to the description under Operand Issue Section. The instruction section examines the error/exception inputs to determine the type of interrupt. It then addresses the appropriate interrupt-handling routine in monitor mode.

OPERAND ISSUE SECTION

The operand issue section consists of the process state and processor state registers, which are located in the 64-word register file or throughout the CP hardware as various live registers.

The 64-word register file contains the operating registers for CYBER 170 State and Virtual State instructions. The operating registers are A, B, and X for CYBER 170 State, and A and X for Virtual State. The register file also holds other exchange package information and provides holding registers for intermediate results.

The live registers contain control information for various CYBER 170 State and Virtual State CP operations. The CP uses the constant output of the live registers during on-line operations. The live registers give the CP access to exchange package information which otherwise would have to be obtained from the register file.

Some of the live registers are writable under microprogram control. Copies maintained in the register file may also be read. These registers typically contain control information rather than data. Normally they are loaded from an exchange package at the same time that exchange package enters the register file from central memory. Other live registers are read-only registers under microprogram control. The contents of these registers change as a result of changes in the CP hardware environment. These changes typically impact system operation and require monitoring.

All register-file registers and live registers are either process state or processor state registers. This distinction arises because the state of the process and the state of the processor characterize CP operation. The contents of the process state registers can be written into memory as a Virtual State exchange package for either a Virtual State process or a CYBER 170 State process. The respective exchange packages are shown in figures I-2-2 and I-2-3. For detailed information on exchange packages and state-switching operations, refer to Interstate Programming in volume II, section 2, listed in the System Publication Index in the preface.

The principal registers of each category are described in the following paragraphs. For detailed functional descriptions of the remaining registers, refer to CP Registers in volume II, section 2, also listed in the System Publication Index.

Process State Registers

The process state registers relate to a specific Virtual State process executing in the CP. Various process state registers also support CYBER 170 State operation. The exchange package for each process contains the step-by-step operating register contents as directed by that process's execution. In addition, the exchange package holds other detailed process state information such that the CP may dynamically switch between exchange packages (processes) while preserving process integrity.

When a process executes in the CP, its exchange package resides in the process state registers. When a process awaits execution, its exchange package resides in central memory. The process state registers include the following:

- Operating registers
  - Sixteen 48-bit address (A) registers, numbered A0 through AF
  - Sixteen 64-bit operand (X) registers, numbered X0 through XF
- Interrupt-handling registers
  - Monitor condition register (MCR) and monitor mask register (MMR), (16 bits each)
  - User condition register (UCR) and user mask register (UMR), (16 bits each)
- Additional process state registers

Refer to table I-2-1 for a complete listing of the process state registers. The principal process state registers are described in this section. The remaining process state registers are described under CP Registers in volume II, section 2, listed in the System Publication Index.

VIRTUAL STATE EXCHANGE PACKAGE

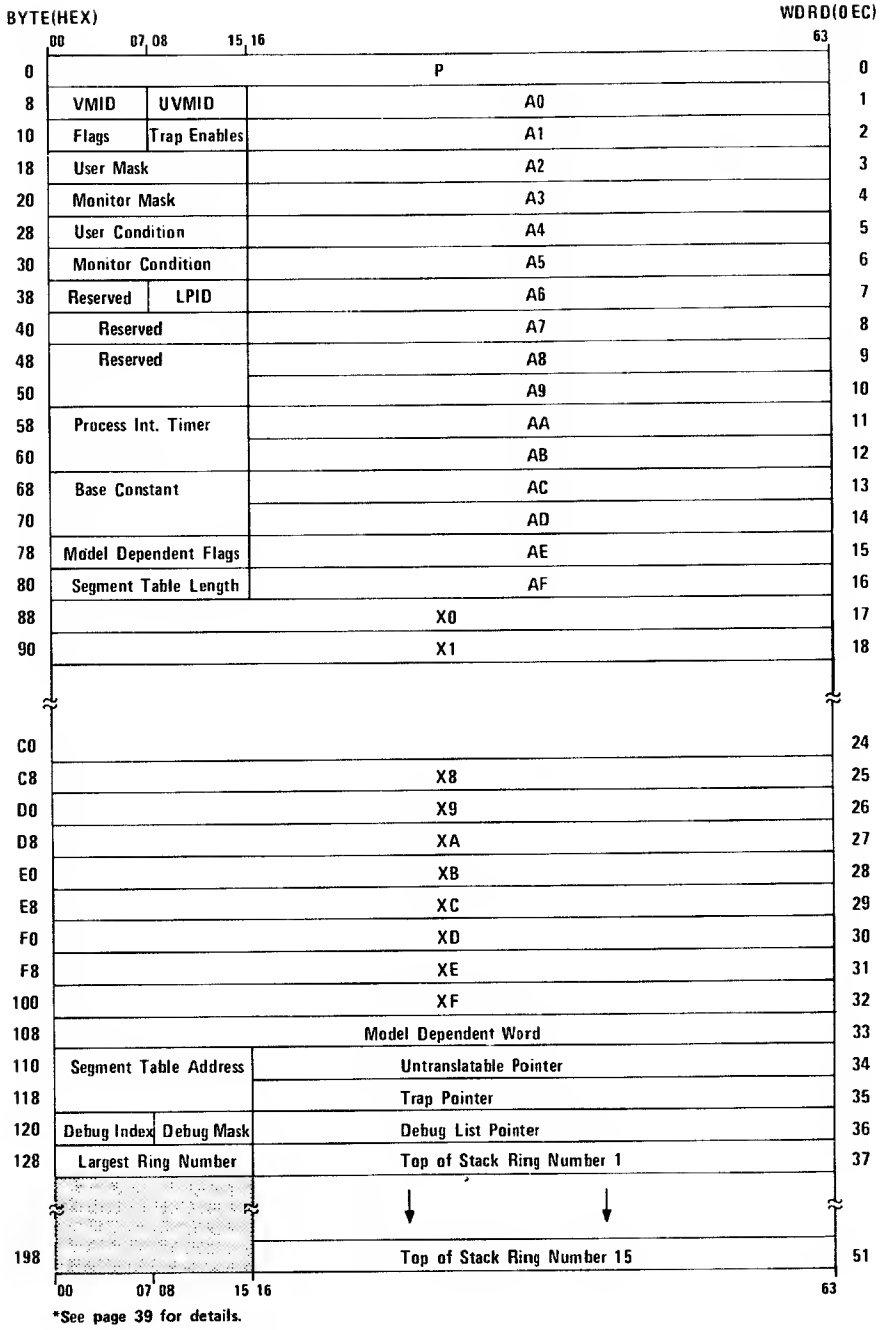
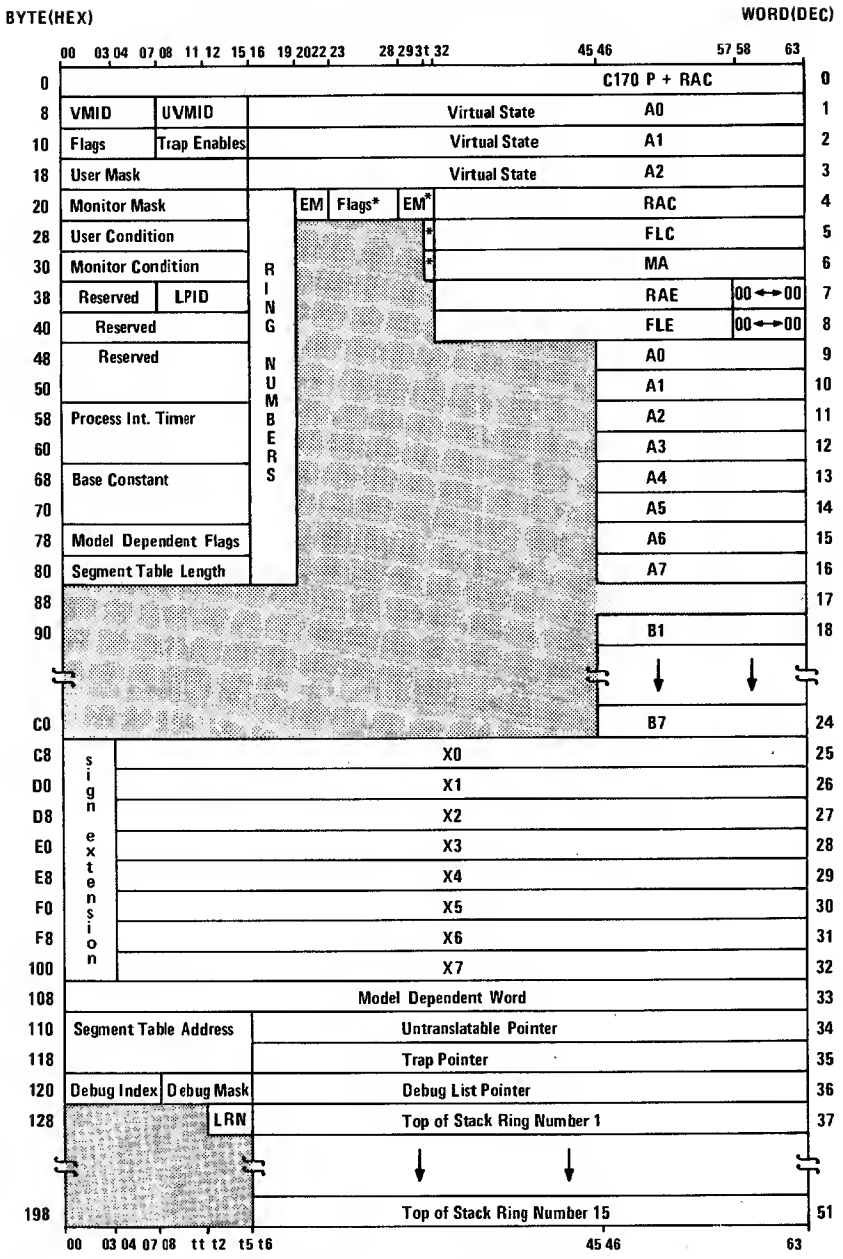


Figure I-2-2. Virtual State Exchange Package

INTERSTATE EXCHANGE PACKAGE



* Flags:

Bit 23, Word 4	UEM Enable Flag
Bit 24, Word 4	Expanded Addressing Select Flag (ESM Mode Flag)
Bit 25, Word 4	Extended Block Copy Flag
Bit 26, Word 4	Software Flag
Bit 27, Word 4	Instruction Stack Purge Flag
Bit 28, Word 4	Software Flag
Bit 29, Word 4	EM-Indefinite Operand
Bit 30, Word 4	EM-Infinite Operand
Bit 31, Word 4	EM-Address out of Range
Bit 31, Word 5	C170 Monitor Flag
Bit 31, Word 6	Exit Mode Halt

Figure I-2-3. Interstate Exchange Package

Table I-2-1. Process State Registers

Register Name	Number of Bits	Register Name	Number of Bits
Address (AO-AF)	48	Process interval timer (PIT)	32
Base constant (BC)	32	Program address (P)	64
Debug index (DI)	6	Segment table address (STA)	32
Debug list pointer (DLP)	48	Segment table length (STL)	12
Debug mask (DM)	7	Top-of-stack pointer (15)	48
Flags		Trap enables (TE)	2
Critical frame (CFF)	1	Trap pointer (TP)	48
On condition (OCF)	1	Untranslatable pointer (UTP)	48
Process not damaged (PND)	1	Untranslatable virtual machine identifier (UVMID)	4
Largest ring number (LRN)	4	User condition (UCR)	16
Last processor identification (LPID)	8	User mask (UM)	16
Monitor condition (MCR)	16	Virtual machine identifier (VMID)	4
Monitor mask (MMR)	16		
Operand (XO-XF)	64		

Operating Registers

The operating registers consist of the address (A) and operand (X) registers, which minimize memory references for arithmetic operands and results.

The time that an exchange package spends in CP hardware is called an instruction interval. During this interval, the operating register contents can be changed by CP instructions (the other process state registers change only as a result of an exchange jump, copy-to-state-register instruction, or branch-on-condition-register instruction).

Address (A) Registers - The A registers are primarily CM operand address registers which contain process virtual addresses (PVAs).

Operand (X) Registers - The X registers are primarily data-handling registers for computation. Depending on the operation, the registers contain a logical quantity, a signed binary integer, or a signed floating-point number. Operands and results transfer between CM and the X registers.

Interrupt-Handling Registers

The monitor condition and user condition registers as well as the associated monitor mask and user mask registers provide the CP interrupt structure. These registers detect interrupt conditions which cause any of the following CP responses:

Exchange — An exchange interrupt switches CP execution to a monitor mode interrupt-handling routine after an inter-mode exchange occurs. An exchange can only be caused by an interrupt condition occurring in job mode. Monitor (system) conditions cause nearly all exchange interrupts; exceptions are user (process) conditions that occur with traps disabled.

Halt — The CP halts and the IOU takes steps to resolve the problem.

Stack — The CP records the condition but temporarily defers processing of the interrupt.

Trap — A trap interrupt switches CP execution to another section of code in the same address space as the executing process. It does not cause an exchange of processes and may occur in both job and monitor modes. A trap is triggered by the occurrence of a certain process interrupt condition. The trapped-to code executes a specific routine that resolves the interrupt and returns control to the process.

These responses to MCR and UCR interrupts depend on whether traps are enabled or disabled and the current operating mode at the time of the interrupt.

The CP usually runs with traps enabled. Traps are disabled to keep them from occurring when certain sections of code are executing. In monitor mode, traps are disabled to prevent interrupts caused by the operating system or peripheral devices. In job mode, traps are disabled to prevent trap interrupts from interrupting the execution of job mode trap-handling routines. However, traps caused by monitor conditions can interrupt these trap-handling routines even if traps are disabled.

CP interrupt responses characteristic of the operating mode are described under the headings Monitor Condition Register (MCR) and User Condition Register (UCR) later in this section.

Monitor Condition Register (MCR) - The MCR provides the CP interrupt structure for interrupt conditions which must be serviced by monitor mode. The MCR contains 16 bits, each of which records a different interrupt condition in the CP. The MCR conditions are of higher priority and are processed before the UCR conditions. The MCR conditions include:

- Hardware errors
- Major software errors
- Addressing and security errors in CP or CM
- Page faults

The specific MCR conditions and corresponding CP responses are listed in table I-2-2, and described in detail under CP Interrupts in volume II, section 2, listed in the System Publication Index in the preface.

An MCR bit set in job mode with traps enabled or disabled causes an exchange to a monitor mode to execute an interrupt-handling routine. This routine analyzes the error and determines corrective action. An MCR bit set in monitor mode with traps enabled transfers control to a trap-handling routine within monitor mode. An MCR bit set in monitor mode with traps disabled causes the CP to stack the condition or halt, depending on the condition.

Each bit in the MCR has an associated mask bit in the MMR. The 16 mask bits allow selective processing of MCR interrupts.

Monitor Mask Register (MMR) - The MMR contains 16 bits, each of which is a mask bit associated with a specific MCR condition. The mask bits control interrupt response when a corresponding MCR bit sets. At certain times during program execution, it may be necessary to defer processing of specific MCR conditions. For example, when executing in CYBER 170 State, the CYBER 170 State exchange request (MCR bit 53) from Virtual State should not be processed and stacked until execution of the previous CYBER 170 State process terminates.

Table I-2-2. Monitor Condition Register

P RGTR	BIT NUMBER AND DEFINITION			ASSOCIATED MONITOR MASK REGISTER BIT SET				MASK BIT CLEAR
				TRAP ENABLED		TRAP DISABLED		TRAP ENABLED OR DISABLED
				JOB MODE	MONITOR MODE	JOB MODE	MONITOR MODE	JOB OR MONITOR MODE
–	48	Detected Uncorrectable Error	Mon	EXCH	TRAP	EXCH	HALT	HALT
–	49	Unassigned		EXCH	TRAP	EXCH	HALT	HALT
P+	50	Short Warning	Sys	EXCH	TRAP	EXCH	STACK	STACK
P	51	Instruction Specification Error	Mon	EXCH	TRAP	EXCH	HALT	HALT
P	52	Address Specification Error	Mon	EXCH	TRAP	EXCH	HALT	HALT
P+	53	170 Exchange Request	Sys	EXCH	TRAP	EXCH	STACK	STACK
P	54	Access Violation	Mon	EXCH	TRAP	EXCH	HALT	HALT
P	55	Environment Specification Error	Mon	EXCH	TRAP	EXCH	HALT	HALT
	56	External Interrupt	Sys	EXCH	TRAP	EXCH	STACK	STACK
	57	Page Table Search Without Find	Mon	EXCH	TRAP	EXCH	HALT	HALT
	58	System Call	Status - This bit is a flag only and does not cause any hardware action.					
P+	59	System Interval Timer	Sys	EXCH	TRAP	EXCH	STACK	STACK
P/P+*	60	Invalid Segment/Ring Number Zero	Mon	EXCH	TRAP	EXCH	HALT	HALT
P	61	Outward Call/Inward Return	Mon	EXCH	TRAP	EXCH	HALT	HALT
P+	62	Soft Error Log	Sys	EXCH	TRAP	EXCH	STACK	STACK
–	63	Trap Exception	Status - This bit is a flag only and does not cause any hardware action.					

* P, unless P+ for RNO on loads

All MMR bits are set if no masking of interrupts is desired. Clearing an MMR bit masks its corresponding interrupt condition in the MCR. The CP Copy-to-State-Register instructions are used to set and clear specific MMR bits. Refer to volume II, section 1, listed in the System Publication Index in the preface.

The MCR is normally all zeros. An interrupt condition sets the appropriate MCR bit. The CP performs a logical product (AND) of the 16 MCR and MMR bits. Refer to figure I-2-4. If the corresponding MMR bit is set (unmasked), an interrupt occurs. If the corresponding MMR bit is clear (masked), the CP either halts or stacks the condition and processes the interrupt later when the mask bit sets.

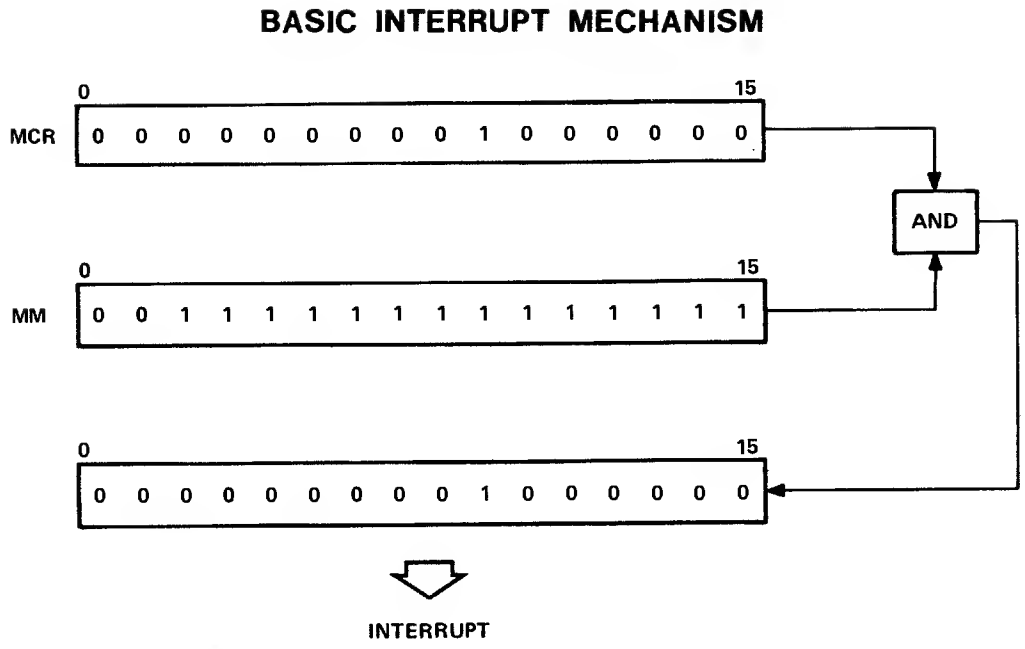


Figure I-2-4. MCR and UCR Interrupt Mechanism

User Condition Register (UCR) - The UCR provides the CP interrupt structure for conditions which relate primarily to instruction execution, and which do not require monitor mode intervention. The UCR contains 16 bits which, like the MCR, record different interrupt conditions in the CP. The UCR conditions include:

- Execution errors
- Arithmetic errors
- Invalid data
- Invalid instructions

The specific UCR conditions are listed in table I-2-3 and described in detail under CP Interrupts in volume II, section 2, listed in the System Publication Index in the preface.

A UCR bit set in job or monitor mode with traps enabled causes a trap to a trap-handling routine within the address space of the process. A UCR bit set in job or monitor mode with traps disabled generally causes the CP to stack the condition. For the monitor-type conditions grouped within the UCR, an interrupt in job mode causes an exchange to monitor mode to execute an interrupt-handling routine, and an interrupt in monitor mode halts the CP.

Each bit in the UCR has an associated mask bit in the (UMR). The 16 mask bits allow selective processing of UCR interrupts.

Table I-2-3. User Condition Register

P RGTR	BIT NUMBER AND DEFINITION		ASSOCIATED USER MASK REGISTER BIT SET				MASK BIT CLEAR
			TRAP ENABLED		TRAP DISABLED		TRAP ENABLED OR DISABLED
			JOB MODE	MONITOR MODE	JOB MODE	MONITOR MODE	JOB OR MONITOR MODE
P	48 Privileged Instruction Fault	Mon	TRAP	TRAP	EXCH	HALT	These mask bits are permanently set.
P	49 Unimplemented Instruction	Mon	TRAP	TRAP	EXCH	HALT	
P	50 Free Flag	User	TRAP	TRAP	STACK	STACK	
P+	51 Process Interval Timer	User	TRAP	TRAP	STACK	STACK	
P	52 Inter-ring Pop	Mon	TRAP	TRAP	EXCH	HALT	
P	53 Critical Frame Flag	Mon	TRAP	TRAP	EXCH	HALT	
–	54 Unassigned	User	–	–	–	–	
P	55 Divide Fault	User	TRAP	TRAP	STACK	STACK	STACK
P	56 Debug	User	TRAP	TRAP	STACK	STACK	STACK
P	57 Arithmetic Overflow	User	TRAP	TRAP	STACK	STACK	STACK
P+	58 Exponent Overflow	User	TRAP	TRAP	STACK	STACK	STACK
P+	59 Exponent Underflow	User	TRAP	TRAP	STACK	STACK	STACK
P+	60 F. P. Loss of Significance	User	TRAP	TRAP	STACK	STACK	STACK
P	61 F. P. Indefinite	User	TRAP	TRAP	STACK	STACK	STACK
P	62 Arithmetic Loss of Significance	User	TRAP	TRAP	STACK	STACK	STACK
P	63 Invalid BDP Data	User	TRAP	TRAP	STACK	STACK	STACK

User Mask Register (UMR) - The UMR contains 16 bits which, like the MMR, are mask bits associated with specific UCR conditions. The mask bits control interrupt action when a corresponding UCR bit sets. At certain times during program execution, it may be necessary or desirable to defer processing of specific UCR conditions. For example, when testing and debugging a new program, it may be desirable to disable interrupts caused by out-of-range arithmetic results, UCR bit 57.

All UMR bits are set if no masking of interrupts is desired. Clearing a UMR bit masks its corresponding interrupt condition in the UCR. The CP copy instruction is used to set/clear specific UMR bits. Refer to CP Copy Instructions in volume II, section 1, listed in the System Publication Index in the preface.

The UCR is normally all zeros. An interrupt condition sets the appropriate UCR bit. The CP performs a logical product (AND) of the 16 UCR and UMR bits. Refer to figure I-2-4. If the corresponding UMR bit is set (unmasked), an interrupt occurs. If the UMR bit is clear (masked), the CP stacks the condition and processes the interrupt later when the mask bit sets. Certain UCR conditions, shown in table I-2-1, require direct attention and cannot be masked.

Additional Process State Registers - The remaining process state registers are accessible to the CP and the PPs. Several of these registers are used by software for performance monitoring or program debugging. Registers also exist to assist the translation from a virtual address to an address used to access data from central memory. These registers are listed in table I-2-3, and described in detail under CP Registers in volume II, section 2, listed in the System Publication Index.

Processor State Registers

The processor state registers contain information about the state of the CP hardware rather than a unique process. This group of registers comprises maintenance registers and other various processor state registers. Refer to table I-2-4.

The maintenance registers provide information about the condition of CP hardware. In some cases, they can be set to force faults in the CP to verify the integrity of the fault detection hardware. The other processor state registers contain pointers/parameters of tables and exchange packages in CM. All processor state registers are accessible to the CP and to the PPs. These registers change only as a result of an exchange jump, if at all. Some may also change under monitor-mode control.

The principal processor state registers are described in this section. The remaining processor state registers are described under CP Registers in volume II, section 2, listed in the System Publication Index in the preface.

Table I-2-4. Processor State Registers

Register Name	Number of Bits	Register Name	Number of Bits
Dependent Environment Control (DEC)	32	Processor Fault Status (PFS)	*
Element Identification (EI)	32	Processor Identification (PI)	8
Job Process State (JPS)	32	Processor Test Mode (PTM)	*
Monitor Process State (MPS)	32	Status Summary (SS)	6
Options Installed (OI)	64	System Interval Timer (SIT)	32
Page Size Mask (PSM)	7	Virtual Machine Capability List (VMCL)	16
Page Table Address (PTA)	32		
Page Table Length (PTL)	8		

* Processor model-dependent. Refer to CP Registers in volume II, section 2, listed in the System Publication Index.

Job Process State (JPS) Register

The JPS register holds the real memory address of the first entry in an exchange package in CM. This address indicates where a job's process state registers are:

- Stored in CM in a job-to-monitor mode exchange
- Retrieved from CM in a monitor-to-job mode exchange

The JPS register works in tandem with the monitor process state (MPS) register in a Virtual State exchange operation. In an exchange from monitor mode to job mode, the CP stores the environment of the monitor mode process in an exchange package at MPS. Refer to figure I-2-5. The CP then initiates the process whose exchange package is located at JPS. This monitor-to-job mode exchange is initiated by an exchange instruction from within the monitor mode process.

In an exchange from job mode to monitor mode, the CP stores the environment of the job mode process in an exchange package at JPS (figure I-2-5). The CP then initiates the process whose exchange package is located at MPS. This job-to-monitor mode exchange is initiated either from an exchange instruction or the occurrence of some interrupt condition requiring monitor mode intervention.

Monitor Process State (MPS) Register

The MPS register holds the real memory address that points to the first entry in an exchange package in central memory. This address indicates where a job's process state registers are:

- Stored in CM in a monitor-to-job mode exchange
- Retrieved from CM in a job-to-monitor mode exchange

The MPS register works in tandem with the JPS register in a Virtual State exchange operation. Refer to Job Process State Register, and to figure I-2-5.

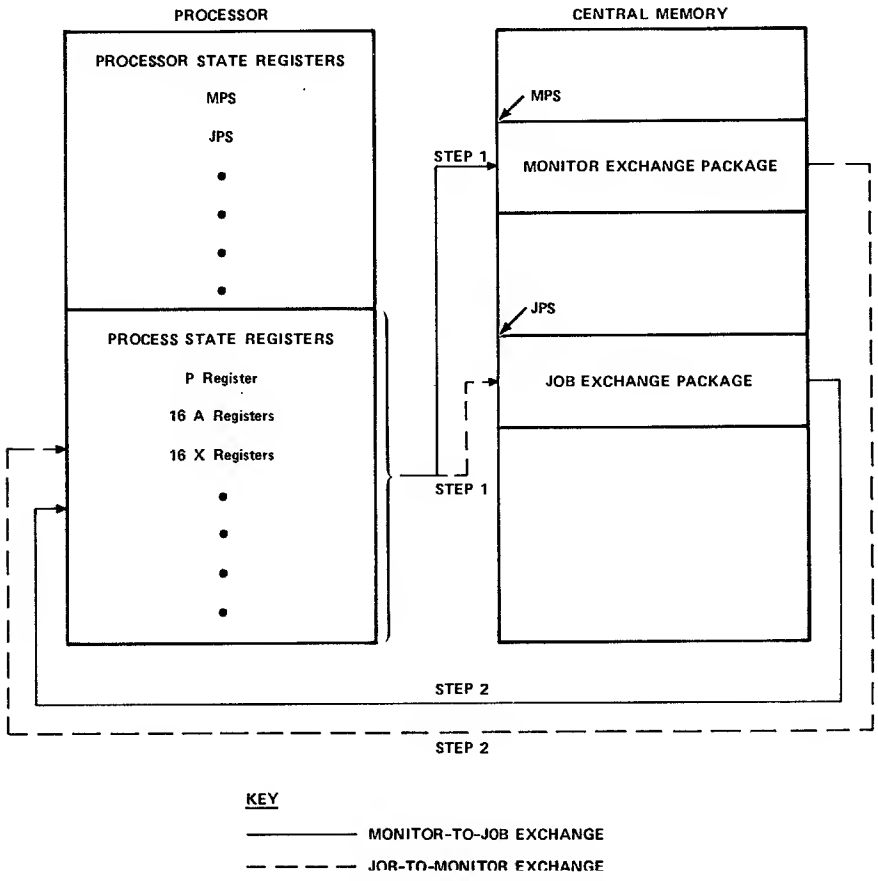


Figure I-2-5. JPS and MPS Functions in Exchange Operation

EXECUTION SECTION

The CP execution section consists of the arithmetic and logical network (ALN). The ALN contains hardware required to execute all floating-point (FP) and integer add, subtract, multiply, and divide instructions found in the CYBER 170 State and Virtual State instruction sets. The ALN also processes CYBER 170 State and Virtual State shift and Boolean instructions, as well as CYBER 170 State pack/unpack, normalize, and population count instructions.

The ALN plays a role in branch instructions by evaluating branch conditions. It performs operations during most other CYBER 170 State and Virtual State instructions, passing operands between register file locations, comparing operands, or completing other required arithmetic steps.

The ALN performs operations on values supplied by the operand issue section or, in some cases, the business data processing (BDP) section. The BDP section performs most BDP operations independently. All results return to the operand issue section except for exception conditions, which are sent to the instruction section to initiate interrupt handling.

The ALN consists of a general network and a multiply/divide network.

General Network

The general network adds and subtracts integers and FP coefficients, and performs exponent arithmetic associated with all FP operations. In Addition, the general network performs Boolean, FP normalize, shift, and conditional branch test functions.

The integers are 60 bits in CYBER 170 State and 64 bits in Virtual State. FP operations use data in the form of an exponent, an exponent bias, a 48-bit (single precision) or 96-bit (double precision) coefficient, and a coefficient bias.

Negative CYBER 170 State operands typically are represented in ones-complement form, with negative Virtual State operands represented in twos-complement form. Operands may be 18 or 60 bits for CYBER 170 State operations; 32 or 64 bits for Virtual State operations.

Multiply/Divide Network

The multiply/divide network forms Virtual State integer products and FP product coefficients for both CYBER 170 State and Virtual State. In the former case, the product may be represented as a 32-bit or 64-bit Virtual State signed integer. In the latter case, the product may be represented as a 48-bit signed coefficient (CYBER 170 State FP operations), or a 48- or 96-bit coefficient in signed-magnitude form (Virtual State FP operations).

The network performs divide operations in which the coefficient of the dividend and divisor operands may be represented as a 32- or 64-bit signed integer (Virtual State only), or in one of the FP coefficient formats described in volume II, section 2, listed in the System Publication Index in the preface.

SEGMENT MAP

Segment map contains the hardware to translate process virtual addresses (PVAs) to system virtual addresses (SVAs). During this translation, segment map also performs the security tests for addressing CM. To reduce CM access time, segment map contains up to 32 of the most recently used segment descriptors from the segment descriptor table.

Segment Map Address Translation

For the PVA-to-SVA translation, segment map obtains PVAs from A registers (for load and store instructions) and the P register (for addressing instructions). Segment map performs the translation by changing the user's segment number to an active segment identifier (ASID) which uniquely identifies each active segment on a system-wide basis. The ASID joins the byte number from the PVA to form the SVA which is sent to a cache in local memory. For further information, refer to Virtual Memory later in this section, or to Virtual Memory Programming in volume II, section 2, listed in the System Publication Index in the preface.

Segment Map Access Validation

The security mechanism in segment map provides controlled access to all code and data. This protects the operating system from users, users from each other, and users from the operating system. The basic element of protection is the user's address space, which is the set of addresses the operating system assigns to an executing process. This address space is defined by the set of segment descriptors in the operating system-maintained segment descriptor table (SDT). Each segment descriptor defines the security protection features for one segment. The protection features consist of security ring tests, key and lock tests, and read/write/execute privilege validity tests. For every CM access attempted, all of these tests must be successful. For further information, refer to Virtual Memory Programming in volume II, section 2, listed in the System Publication Index.

CM Access Via Segment Map

As previously stated, segment map contains up to 32 of the most recently used 64-bit segment descriptors from the segment descriptor table (SDT) in CM. The CP tests whether the requested segment descriptor is in segment map on every memory reference. In this test, the CP uses the lower 4 bits of the segment number to index 2 of the 32 most recently used segment descriptors. It then compares the requested segment number's upper 8 bits to the indexed segments' upper 8 bits. If they match, segment map sends the ASID in this segment descriptor to local memory for the SVA-to-RMA translation. If the segment descriptor is not available in segment map, the CP obtains it from the SDT.

LOCAL MEMORY

Local memory contains the hardware to translate SVAs to RMAs. Two buffer memories (also in local memory) reduce CM access time. The buffer memories are cache memory and page map memory. Cache memory has 2048 or 4096 words (32K bytes) of the most recently used entries in system virtual memory. Page map has up to 128 of the most recently used page descriptors from the system page table.

The CP simultaneously tests cache memory and page map for presence of the requested SVA. If a cache memory hit occurs, no further action is required because the CP reads the desired data from cache memory (refer to CM Access Via Cache Memory). If the desired data is not in cache memory, the page map test is relevant (refer to CM Access Via Page Map). If a page map hit occurs, the SVA-to-RMA translation occurs and the CP reads a four-word block of data from CM (refer to Page Map Address Translation). If no page map hit occurs, the CP initiates a search of the system page table (SPT). If the page is not listed in the SPT, the CP retrieves it from external mass storage.

CM Access Via Cache Memory

Cache memory is a high-speed buffer memory transparent to the user. Since it is a smaller and faster memory than CM, cache memory effectively reduces CM access time by eliminating unnecessary CM references.

On first reference to a word in CM, the CP rapidly reads a block of four CM words containing the requested word into cache memory. On the subsequent reading of any word in this block, CM need not be accessed because these words are in cache memory. The probability of a cache hit is well over 90 percent for most data processing applications because:

- The same data is often read more than once
- The CP may repeatedly execute a loop of instruction words
- There is data lookahead from writing successive blocks of instructions or data in cache when CM words are read sequentially

Cache contains 2048 words (16K bytes) of the most recently used entries in system virtual memory. An optional 4096-word (32K-byte) cache is available.

The CP tests whether the requested entry is in cache on every memory reference. The CP then compares the requested entry to the four entries in each set. If no cache hit occurs, the CP reads a set of four new words containing the requested entry into the indexed position. The CP uses the entry's lower 5 bits to select the requested word, and the byte within that word, from the new set.

Virtual State monitor mode may designate certain virtual memory segments as cache bypass segments. The CP never reads this information into cache memory.

The CP uses the lower 8 bits of the requested entry (address) to index one of the two (16K-byte cache) or four (32K-byte cache) sets of four-word blocks in cache. The 16K-byte cache contains 128 sets; the 32K-byte cache contains 256 sets.

CM Access Via Page Map

If the simultaneous test of cache memory and page map does not produce a cache hit, the page map test is relevant. As previously stated, page map contains up to 128 of the most recently used page descriptors from the software-managed system page table (SPT) in CM. The CP tests whether the requested page descriptor is in page map. In this test, the CP uses the lower 5 bits of the segment/page identifier (SPID) to index the 32 most recently used page descriptors in the system page table (SPT). If they match (a page map hit), page map translates the SVA to an RMA and the CP reads a four-word block of data from CM. If no page map hit occurs, the CP initiates a search of the system page table and obtains the page descriptor.

Page Map Address Translation

If no cache or page map hit occurs, page map translates the SVA from segment map to an RMA. Page map performs the translation by hashing (exclusive OR) the active segment identifier (ASID) and page number to form the SPID. The SPID provides an index into the SPT. From the indexed location, the CP searches the SPT to find the page descriptor with the required SPID. Along with the SPID, the page descriptor contains a page frame address (PFA), which is the starting address of each page currently in CM. System software uses the PFA to generate the RMA where the requested data resides. For further information, refer to Virtual Memory Programming in volume II, section 2, listed in the System Publication Index in the preface.

ADDRESSING SECTION

The addressing section has the following functions:

- Forms the byte number portion of the SVA sent to local memory.
- Performs address arithmetic and data manipulation for loading and storing data in CM. The load and store instructions involve transferring a single bit, byte stream, word, or multiple words between register file locations (in the operand issue section) and CM locations.
- Performs the Test and Set Bit instruction, which transfers one bit from a CM location to a specified, constant bit position in an X register.
- Contains address streaming logic for BDP instructions. This includes disassembling 64-bit words from local memory into 8-bit bytes for the BDP section and assembling BDP bytes into 64-bit words for local memory.

BUSINESS DATA PROCESSING (BDP) SECTION

The BDP section executes BDP instructions that operate on CM data fields up to 256 bytes in length. Although the BDP section performs most BDP operations independently, for some operations it may require processing assistance from the ALN in the execution section. BDP operations use any of 16 data types which are organized as follows:

- Packed decimal (4)
- Unpacked decimal (5)
- Binary (2)
- Alphanumeric (1)

In many cases, the data types may be freely mixed as the hardware performs the type translations required for various BDP operations. Refer to BDP Programming in volume II, section 2, for descriptions of the data types.

BDP instructions reference BDP data via data descriptors, which are in the main instruction stream and which contain information about the location, size, and type of data. The BDP data descriptors also specify two data fields in CM: the source field and the destination field. The former modifies, replaces, or compares to the latter.

MAINTENANCE ACCESS CONTROL (MAC)

The MAC performs initialization and maintenance operations in the CP. These operations, controlled by a dedicated peripheral processor in the IOU, include the following:

- Initialize registers, controls, and memories
- Read and write CP-resident registers and memories
- Monitor and record CP error/status information
- Verify error detection and correction hardware
- Reconfigure CP

The IOU sends codes to MAC to indicate the operations to be performed and the affected memory device, registers, and addresses. Certain MAC registers contain CP and CM fault isolation information. Other registers control the internal configuration and operation of the CP and verify that errors are properly reported and recorded. For more detailed information, refer to Maintenance Channel Programming in volume II, section 2, listed in the System Publication Index in the preface.

CENTRAL MEMORY (CM)

The CM performs the following functions:

- Eight memory banks store from 512K to 2048K 64-bit words with 8-bit SECDED codes.
- Multiple ports make CM accessible to the CP and every PP.
- CM bounds register limits writes to CM from any or all ports.
- SECDED, stored with each word, checks circuits, corrects single-bit errors, and detects double-bit errors.
- Maintenance channel interface gives a PP in the IOU access to the CM maintenance registers for system initialization, corrective action, error reporting and diagnostics, and for setting the port bounds register.

ADDRESS FORMAT

Figure I-2-6 illustrates the address format.

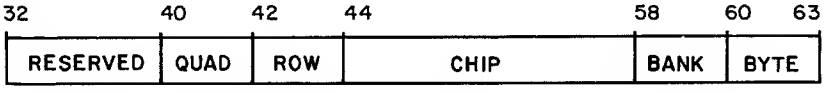


Figure I-2-6. Address Format

The following list defines the address fields.

- Byte select specifies one of eight bytes within a word. When used, the CP codes these 3 bits to 8 mark bits before accessing memory. These 3 bits are not part of the address proper.
- Bank select specifies one of eight banks. Since the bank address is the lower 3 bits of the storage address, sequential addressing results in a phased-bank operation which allows a maximum data transfer rate of one word each clock period.
- Chip address specifies the address of one word in a chip for the selected bank.
- Row select specifies one of four word rows in a quadrant.
- Quadrant select specifies one of four quadrants.

CM ACCESS AND CYCLE TIMES

The CM access time for a read operation is 672 nanoseconds.

Cycle time for a read or write operation is 8 clock periods (448 nanoseconds). Cycle time for a partial write (read/write/modify) is 16 clock periods (896 nanoseconds).

The CM bounds register limits CM write access to an area between two addresses specified in this register. The CM bounds register is set through the maintenance channel. Refer to Maintenance Channel Programming in volume II, section 2, listed in the System Publication Index in the preface.

CM PORTS AND PRIORITIES

A priority network resolves port access conflicts on a rotating basis, preventing long term lockout of any port. In case of simultaneous requests, the CP has priority. Refer to table I-2-5 for maximum request lockout time.

Table I-2-5. Port Priority

Port	Maximum Request Lockout Time in Bank Cycles	
	Read or Write	Partial Write
Port 0	4	7
Port 1	5	9
Port 2	6	11
Port 3	7	13

1 bank cycle = 8 clock periods = 448 nanoseconds

SECDED

The SECDED logic corrects single-bit errors during a CM read operation, permitting unimpeded computer operation. The SECDED logic prepares for the error correction by generating error correction code (ECC) bits for each data word and by storing these ECC bits in CM with the data word during the CM write. Table I-2-6 lists the hexadecimal codes for all the combinations of syndrome bits with the number of the data bit assigned each code or a note categorizing each code. Then, during a CM read, CM performs the following SECDED sequence:

1. Read one CM word and generate new ECC bits for data portion of CM word.
2. Compare new ECC bits with CM word ECC bits.
3. If old and new ECC bits match, no error exists. Send data to requesting unit.
4. If bits do not match, generate syndrome bits from result of ECC compare.
5. Decode syndrome bits to determine if single- or multiple-bit failure.
6. If single-bit failure, correct by inverting failing bit in data word. Send corrected word to requesting unit. (Also sets Soft Error Log bit in MCR.)
7. If multiple-bit or other uncorrectable error, send uncorrectable error response code to CP or IOU. (Also sets Detected Uncorrectable Error bit in MCR.) A PP in the IOU may then analyze the syndrome bits using the maintenance channel.

Certain CM registers assist in SECDED analysis. The Corrected Error Log (CEL) register displays details of the first corrected error and the Uncorrectable Error Log (UEL) register displays details of the first two uncorrected errors. For detailed descriptions of these registers, refer to CM Registers in volume II, section 2, listed in the System Publication Index in the preface. The syndrome bits may be analyzed through the maintenance channel.

CM BOUNDS REGISTER

The CM bounds register limits the write access to CM from specified ports. The ports are limited to the area between an upper and lower bound as specified in the CM bounds register. Bits in byte 0 specify the port(s) from which the write access is limited. The CM bounds register is set through the maintenance channel. For further information, refer to Maintenance Channel Programming in volume II, section 2, listed in the System Publication Index.

CM MAINTENANCE REGISTERS

The CM contains maintenance registers which hold memory status and error information. These registers are accessible through the maintenance channel. Table I-2-7 lists the CM maintenance registers. For detailed decriptions of these registers refer to CM Registers in volume II, section 2, listed in the System Publication Index.

Table I-2-6. SECDED Syndrome Codes/Corrected Bits

Code	Bit	Code	Bit	Code	Bit	Code	Bit	Code	Bit	Code	Bit	Code	Bit	Code	Bit
00	(7)	20	66 (2)	40	65 (2)	60	(3)	80	64 (2)	A0	(3)	C0	0/1 (6)	E0	32 (1)
01	71 (2)	21	(3)	41	(3)	61	(4)	81	(3)	A1	(4)	C1	(4)	E1	32 (5)
02	70 (2)	22	(3)	42	(3)	62	(4)	82	(3)	A2	(4)	C2	(4)	E2	36 (5)
03	6/7 (6)	23	(4)	43	(4)	63	(3)	83	(4)	A3	(3)	C3	(3)	E3	36 (1)
04	69 (2)	24	(3)	44	(3)	64	(4)	84	(3)	A4	(4)	C4	(4)	E4	34 (5)
05	(3)	25	(4)	45	(4)	65	(3)	85	(4)	A5	(3)	C5	(3)	E5	34 (1)
06	(3)	26	(4)	46	(4)	66	(3)	86	(4)	A6	(3)	C6	(3)	E6	38 (1)
07	24 (1)	27	28 (5)	47	26 (5)	67	30 (1)	87	25 (5)	A7	29 (1)	C7	27 (1)	E7	31/38 (5)
08	6B (2)	28	(3)	48	(3)	6B	(4)	88	(3)	A8	(4)	C8	(4)	E8	33 (5)
09	(3)	29	(4)	49	(4)	69	(3)	89	(4)	A9	(3)	C9	(3)	E9	33 (1)
0A	(3)	2A	(4)	4A	(4)	6A	(3)	8A	(4)	AA	(3)	CA	(3)	EA	37 (1)
0B	16 (1)	2B	20 (5)	4B	1B (5)	6B	22 (1)	8B	17 (5)	AB	21 (1)	CB	19 (1)	EB	23/37 (5)
0C	4/5 (6)	2C	(4)	4C	(4)	6C	(3)	8C	(4)	AC	(3)	CC	(3)	EC	35 (1)
0D	8 (1)	2D	12 (5)	4D	10 (5)	6D	14 (1)	8D	9 (5)	AD	13 (1)	CD	11 (1)	ED	15/35 (5)
0E	0 (1)	2E	4 (5)	4E	2 (5)	6E	6 (1)	8E	1 (5)	AE	5 (1)	CE	3 (1)	EE	7/39 (5)
0F	(3)	2F	(4)	4F	(4)	6F	(3)	8F	(4)	AF	(3)	CF	(3)	EF	39 (1)
10	67 (2)	30	2/3 (6)	50	(3)	70	56 (1)	90	(3)	B0	48 (1)	D0	40 (1)	F0	(3)
11	(3)	31	(4)	51	(4)	71	56 (5)	91	(4)	B1	48 (5)	D1	40 (5)	F1	(4)
12	(3)	32	(4)	52	(4)	72	60 (5)	92	(4)	B2	52 (5)	D2	44 (5)	F2	(4)
13	(4)	33	(3)	53	(3)	73	60 (1)	93	(3)	B3	52 (1)	D3	44 (1)	F3	(3)
14	(3)	34	(4)	54	(4)	74	58 (5)	94	(4)	B4	50 (5)	D4	42 (5)	F4	(4)
15	(4)	35	(3)	55	(3)	75	58 (1)	95	(3)	B5	50 (1)	D5	42 (1)	F5	(3)
16	(4)	36	(3)	56	(3)	76	62 (1)	96	(3)	B6	54 (1)	D6	46 (1)	F6	(3)
17	24 (5)	37	28 (1)	57	26 (1)	77	30/62 (5)	97	25 (1)	B7	29/54 (5)	D7	27/46 (5)	F7	(3)
1B	(3)	3B	(4)	58	(4)	78	57 (5)	98	(4)	B8	49 (5)	D8	41 (5)	F8	(4)
19	(4)	39	(3)	59	(3)	79	57 (1)	99	(3)	B9	49 (1)	D9	41 (1)	F9	(3)
1A	(4)	3A	(3)	5A	(3)	7A	61 (1)	9A	(3)	BA	53 (1)	DA	45 (1)	FA	(3)
1B	16 (5)	3B	20 (1)	5B	18 (1)	7B	22/61 (5)	9B	17 (1)	BB	21/53 (4)	DB	19/45 (5)	FB	23 (1)
1C	(4)	3C	(3)	5C	(3)	7C	59 (1)	9C	(3)	BC	51 (1)	DC	43 (1)	FC	(3)
1D	8 (5)	3D	12 (1)	5D	10 (1)	7D	14/59 (5)	9D	9 (1)	BD	13/51 (5)	DD	11/43 (5)	FD	15 (1)
1E	0 (5)	3E	4 (1)	5E	2 (1)	7E	6/63 (5)	9E	1 (1)	BE	5/55 (5)	DE	3/47 (5)	FE	7 (1)
1F	(4)	3F	(3)	5F	(3)	7F	63 (1)	9F	(3)	BF	55 (1)	DF	47 (1)	FF	(3)

Notes:

1 Corrected single-bit error.
2 Syndrome code bit failed (single code bit set).
3 Double error or multiple error (even number of code bits set).
4 Multiple error reported as a single error.
5 Double error or multiple error with indicated bit(s) inverted.
6 Double error or multiple error or forced double error due to a partial write parity error on one of the two bytes.
7 No error detected.

Table I-2-7. CM Maintenance Registers

Register Name	Number of Bits
Bounds Register	64
Corrected Error Log	64
Element Identification (EI)	32
Environment Control (EC)	32
Free-Running Counter	48
Options Installed (OI)	4
Status Summary	6
Uncorrectable Error Log 1	64
Uncorrectable Error Log 2	64

CM RECONFIGURATION

CM reconfiguration permits the computer operator to restructure the CM addresses so that a failing part of CM can be quickly locked out to provide a continuous block of usable CM from address zero. CM reconfiguration is accomplished by setting the switches to manipulate the upper address bits. For further information, refer to the hardware operator's guide listed in the preface.

VIRTUAL MEMORY

Central memory functions as a virtual memory. Virtual memory effectively gives each user a vast, unique address space complete with a copy of the operating system. Actually, most user code and data reside in external mass storage, and virtual memory code sharing enables the apparent duplication of operating system routines such as Fortran compiler, trap handler, and memory manager.

Virtual memory is discussed in the following paragraphs. For information regarding operation of CYBER 170 State (real) memory, refer to the appropriate CYBER 170 State hardware reference manual listed in the preface.

The operating system, with hardware support, manages virtual memory by segmenting virtual memory code and data and mapping this information to real memory pages.

Segments

Segments, shown in figure I-2-7, are the units of virtual memory storage. Each user's executing process operates in a unique virtual address space divided into a number of segments. A maximum of 4096 segments may exist for each process, and over 65,000 segments may exist system-wide for all user processes. Each segment has a capacity of 2000 Megabytes (262 million words). Segment size varies according to the amount of information it contains.

In addition to partitioning the virtual address space, the segment typically provides the natural divisions of code and data in a process. For instance, one segment may hold data files used by the process, another executable code unique to the process, and another a duplicate copy of operating system code shared with other users.

To optimize the use of real memory, operating system segments which reside in user address space (that is, monitor mode interrupt handler) are shared by several users, and thus conceptually exist as multiple copies in virtual memory.

Segments also play an important role in access protection. Each segment is assigned a set of read, write, and execute attributes which characterize the use of that segment by various users. For further information, refer to Access Protection later in this section, or to Virtual Memory Programming in Volume II, section 2, listed in the System Publication Index in the preface.

Pages

Pages, shown in figure I-2-8, are the units of real memory storage. Pages also serve as the common storage unit between virtual and real memory. Although pages exist within segments, hardware carries the pages only as bookkeeping in virtual memory. The virtual-to-real page translation facilitates the efficient use of available CM through demand paging. This translation eliminates the need for programmer-created program overlays.

Page size (the number of memory words per page) is a fixed length selected at system initialization and is constant from one deadstart to the next. Pages may be 2K, 4K, 8K, or 16K bytes in length. The number of pages per segment is proportional to page size.

Central memory comprises physical sections of equal size called page frames. Pages are the same size as page frames; thus, one page of information from external mass storage loads (on demand) into one page frame in CM. A page may or may not reside in CM, depending on its recent use and the competing demand for CM space in retrieving other process's pages from external mass storage. Many more pages exist in virtual memory than in real memory, and pages are brought from virtual into real memory on a demand basis. Data transfers between CM and external mass storage one page at a time.

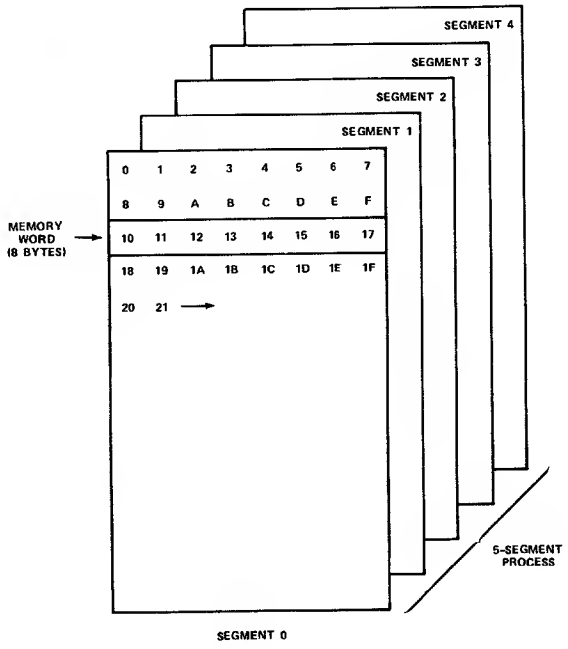


Figure I-2-7. Segments in a Process

Address Translation

CP instructions address memory by way of process virtual addresses (PVAs). A PVA is the only type of address a user sees. To locate the requested memory word, the CP translates a PVA to a real memory address (RMA). Hardware performs the translation in two steps by 1) converting a PVA to a system virtual address (SVA), and 2) converting an SVA to an RMA.

The distinctly different functions performed by the two steps are:

- The PVA-to-SVA translation performs a security check which determines whether the process has the necessary privilege to access the requested code/data.
- The SVA-to-RMA translation performs the memory management task of converting the system-wide virtual address to an RMA, which locates the requested code/data in central memory or external mass storage. In the latter, the operating system retrieves the relevant real memory page from external mass storage into CM.

The operating system maintains tables in memory which make possible the PVA-to-SVA-to-RMA conversion with access protection (refer to figure I-2-9). When translating a PVA to an SVA, the CP uses a software-managed segment descriptor table that describes the unique address space for that segment. When translating an SVA to an RMA, the CP uses the software-managed system page table that contains a page descriptor for each active page in CM.

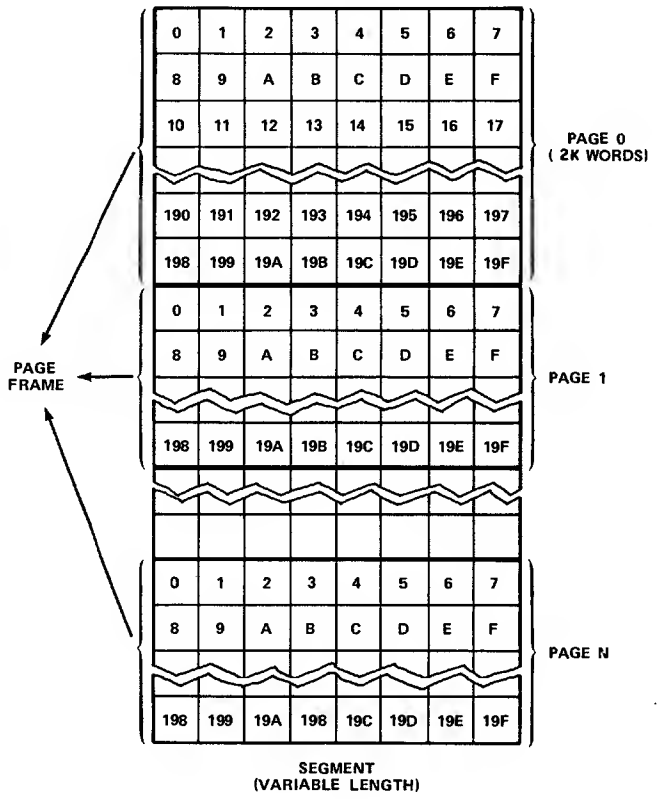


Figure I-2-8. Pages in a Segment

Access Protection

Access protection consists of constraining each user to an address space and preventing unauthorized reading, writing, or executing of code or data outside this address space. As previously mentioned, the nature of virtual memory allows the operating system to exist in the programmer's address space. Access protection mechanisms are therefore necessary to ensure that protection between system code and user code/data occurs at all times.

The operating system sets access requirements and privileges from information on file or by responding to legitimate requests from the user. The user cannot directly change access requirements or increase access privileges as set by the operating system because the operating system lists the access requirements of process segments in user-inaccessible areas in CM and places access privilege information with the program counter and address registers.

When the CP presents an address for translation, hardware tests for proper access privilege. Testing of privilege occurs by way of attributes, rings of protection, and keys and locks. All three tests must be successful, or the CP interrupts the process and performs an exchange to another process. The monitor mode interrupt handler determines appropriate action for the interrupted process.

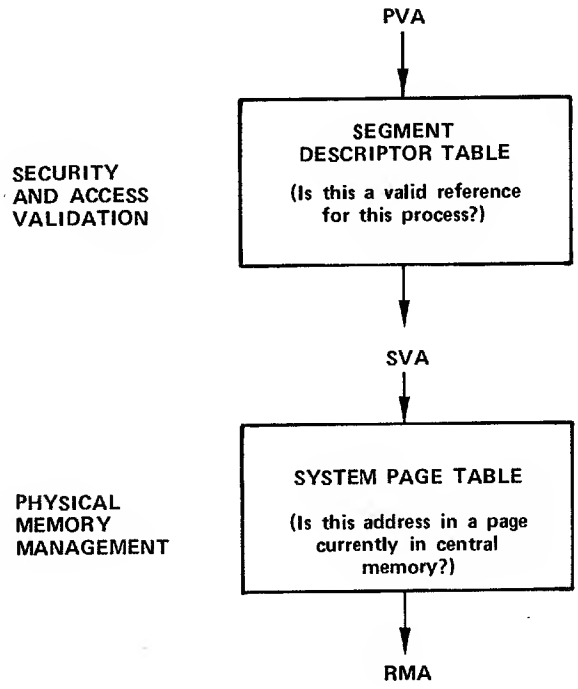


Figure I-2-9. Segment/Page Table Role in Address Translation

Attributes

The operating system assigns a set of read/write/execute attributes to each segment; any combination may be assigned. These attributes characterize the use of that segment by various users. The operating system records the attributes in a segment descriptor table for each process and references them at the PVA level of address translation. It is possible for more than one process to share a segment with each process having different access attributes. This forms the basis for code sharing.

When first referencing a PVA, the CP compares the type of reference (read/write/execute) to the attributes of the segment where the PVA resides. To complete the operation, the reference privilege must match a like segment attribute. For instance, reading a particular data segment cannot occur unless the requested segment has a read attribute.

Rings of Protection

The system virtual address space comprises 15 rings of protection. The rings primarily prevent unauthorized access, although rings typically also separate code and data segments. The 15 rings have a hierarchical organization such that the lower the ring number, the higher the privilege. For instance, operating system segments occupy the lowest rings, and the least privileged user segments occupy the highest rings.

Code and data segments for a process may exist in several rings. When this occurs, the segment is said to reside in a ring bracket. There are four ring brackets associated with each segment. These are read, write, execute, and call brackets. The brackets are an extension of the read/write/execute access attributes. On an initial reference to a PVA, the CP compares the ring bracket of the process attempting the read, write, execute, or call to the ring number of the segment where the PVA exists. To complete the access, the referenced segment must reside in an accessible ring bracket.

Keys and Locks

Keys and locks provide a protection mechanism for segments that reside within the same ring of protection. This function includes:

- The protection of local data used by a particular procedure(s)
- The isolation of competing applications residing in the same ring

The CP associates a lock with each segment and, in general, only grants access to one segment from another if the keys exactly match the locks. Thus, this protection mechanism has no hierarchical significance but depends only on whether the keys and locks match.

INPUT/OUTPUT UNIT

The IOU performs the functions necessary to locate, select, and initialize the external devices connected to the system. The IOU also controls the transfer of data between a selected device and CM as well as performing system maintenance functions.

The IOU contains the following functional areas:

- Peripheral processors (PPs)
- I/O channels
- Maintenance channel
- CM access

PERIPHERAL PROCESSOR

The basic IOU contains 10 PPs and can be expanded to 20 PPs in 5-PP increments. Each PP is a logically independent computer with its own memory. Each 5-PP group is organized into a multiplexing system which allows the PPs to share common hardware for arithmetic, logical, and I/O operations without losing independence. This multiplexing system comprises five ranks of registers termed a barrel. Each rank contains information related to the instruction being executed by one PP.

Each PP can communicate with:

- Other PPs over the I/O channels and through CM
- The CP via CM read and write operations
- The CP (in CYBER 170 State operation) by issuing a CYBER 170 State exchange request to a specific CYBER 170 State exchange package associated with the issuing PP

Each PP can also cause an interrupt condition with the CP operating in either Virtual State or CYBER 170 State.

Each PP executes programs alone or with other PPs to control data transfers between external devices and CM. These programs, called I/O drivers, comprise IOU instructions combined to interact with operating system requests issued through CM. The I/O drivers translate generalized operating system requests into control functions for accessing the external devices and may also perform device scheduling and optimization. The I/O drivers use PP memory as a buffer for the data transfer between external devices and CM to isolate IOU data transfers from variations in CM transfer rate.

DEADSTART

A deadstart sequence allows the IOU to initialize itself. This deadstart sequence is initiated by the DEAD START switch on the deadstart panel or the DEAD START switch on the display station. The panel includes controls for assigning any PP memory (PPM) to PP0. For further information, refer to the hardware operator's guide listed in the preface.

BARREL AND SLOT

The barrel, as shown in figure I-2-10, consists of the A, K, P, Q, and R registers, each of which has five ranks numbered 0 through 4. Information in these registers is transferred from one rank to the next at a uniform 20-megahertz rate, providing a multiplexed system of five PPs, each operating at a 4-megahertz rate. The registers are

stationary while the PPs rotate. For example, rank 4 registers contain PP0, PP1, PP2, PP3, and PP4 in succession, each consuming 50 nanoseconds of the total cycle time of 250 nanoseconds. Since the PP memories operate at a slower rate, independent memory with its own address and data registers is provided for each PP.

Each time data enters the slot, a portion of the instruction for that data is executed. The slot performs tasks such as arithmetic and logic operations, and program address manipulation. Complete execution of an instruction may require the A, K, P, Q, and R register quantities to go more than one trip around the barrel and through the slot.

The PPM may be referenced once each time the PP passes around the barrel and through the slot. During its slot time, the PP may also communicate with CM or any of the I/O channels.

PP REGISTERS

The PP registers consist of the A, K, P, Q, and R registers. The registers and their descriptions follow.

A Register

The 18-bit A register contains one of two operands for arithmetic and logic operations. The content of A may be an arithmetic operand, a CM address or part of a CM address, an I/O function, an I/O data word, or a word count for a block I/O or CM transfer. Various instructions operate on 6, 12, or 18 bits of the A register. Calculation results are always placed in the A register, although some instructions also write the result into PP memory.

When the A register provides the CM address, parity is generated with the address for transmission to memory control. When the A register provides data or function words for I/O activities, channel parity is always generated on 16 bits of A.

At deadstart, the A register is set to 10000_8.

K Register

The 12-bit K register is visible to the programmer through the maintenance channel and the IOU deadstart panel. This register holds the operation code field of an instruction for display and is used for maintenance purposes. When a PP is halted (idled), this register contains all ones.

P Register

The 12-bit P register is the PP program address counter. Also, during block I/O and CM transfers, the P register temporarily contains the PP memory address of the data transfer. At deadstart, the P register is set to 7777_8.

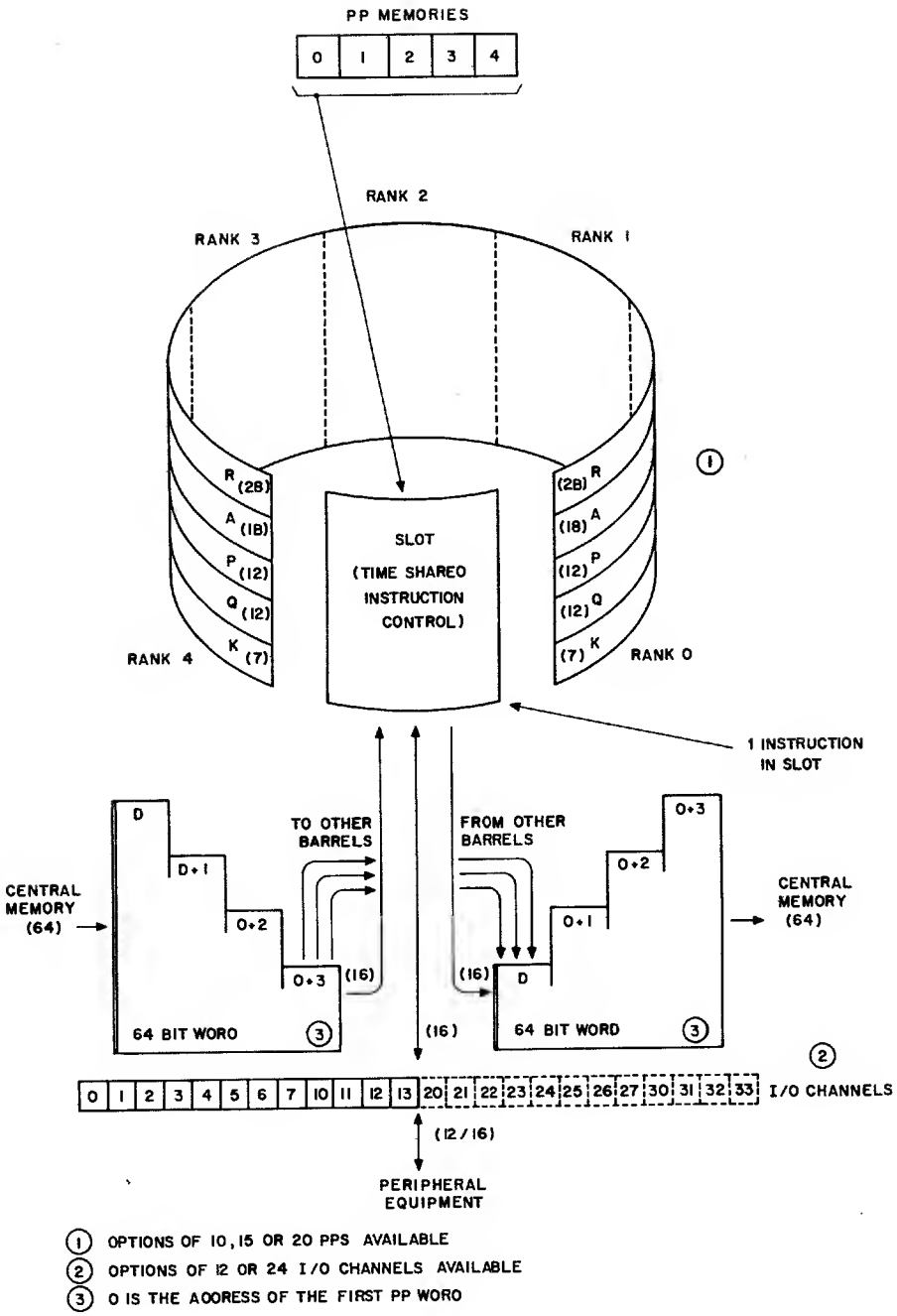


Figure I-2-10. IOU Barrel and Slot

Q Register

The 12-bit Q register may hold the following data:

- Operand address for direct and indirect addressing
- Peripheral address of data used during single-word CM read/write instructions
- Shift count
- Word count for CM block transfers
- Upper 6 bits during constant mode PP instructions
- Target address for relative jump
- Channel number for all I/O and channel instructions

At deadstart, each rank of the Q register is set to a corresponding PP number. Rank 0 is set to PP0, rank 2 is set to PP2, and so on.

R Register

The 28-bit R register, in conjunction with the A register, forms an absolute CM address for CM read/write instruction (refer to Central Memory Access by PPs).

PP NUMBERING

PPs are numbered in octal as follows:

Barrel	PPs
0	00 through 04
1	05 through 11
2	20 through 24
3	25 through 31

The deadstart sequence decodes deadstart panel switches to determine PP numbering within a barrel. The sequence assigns barrel numbers according to the switch settings and, during the first minor cycle after deadstart, loads a zero into the Q register in barrel 0. This defines all the data in that rank of the barrel as belonging to PP0 and since Q is the channel selector, assigns PP0 to channel 0. During the next minor cycle, Q loads with a 1. This defines PP1 and assigns it to channel 1. This process occurs in parallel in all barrels until the IOU assigns each rank of the barrel with a PP number and a channel number. Reassignment can only be done at deadstart.

PP MEMORY

Each PP has an independent 4K word memory; each word contains 16 data bits and 1 parity bit. PP0 must be operational since it reads the deadstart program from the deadstart panel during the deadstart operation. A PP memory reconfiguration feature allows the user to restore IOU operation if the IOU detects a fault in the PP memory normally assigned to PP0.

To reconfigure, the operator assigns a good PP memory to PP0, and the operating system removes the failing PP memory. Operation continues around the failing PP memory, to allow repair during scheduled maintenance. The system must be deadstarted to reconfigure PPs.

I/O CHANNELS

The I/O channels comprise two interfaces. An internal interface allows common hardware and software to control the external devices; an external interface allows communication with the external devices using 12/16-bit data channels or an 8-bit maintenance channel.

The internal interface can transfer a maximum of one word every 250 nanoseconds between two PPs, or between a PP and an external device. This rate can be sustained for the maximum practical channel transfer (4096 words). During transfers between PPs, if the PPs are not in the slot at the same time, the transfer rate is one every 500 nanoseconds.

The external interface contains the interface mechanisms to connect the appropriate channel for an external device. The transfer rate between an I/O device and a PP is a function of the channel type and the maximum data transfer rate of the I/O device.

All PPs communicate with external devices through the independent I/O channels. Each channel may be connected to one or more pieces of external equipment, but only one piece of equipment can use a channel at one time. All channels can be active simultaneously.

DISPLAY STATION CONTROLLER

The display station controller (DSC) is the IOU interface between the PPs and the display station servicing both the keyboard and the cathode-ray tube (CRT). The DSC transmits function words and digital symbol size/position data to the display station and receives digital character codes from the keyboard. It also receives digital symbol codes from the PPs and converts these to analog symbols to the CRT.

REAL-TIME CLOCK

The real-time clock is a 12-bit free-running counter, incrementing at a 1-megahertz rate. It is permanently attached to channel 14_8. This channel may be read at any time since it is active and full flags are always set.

TWO-PORT MULTIPLEXER

The two-port multiplexer provides communication capability between a PP and two attached terminals. It can simultaneously drive the two terminals at different baud rates. One port is reserved for maintenance purposes, and the other port is reserved for future use. The two-port mulptiplexer is permanently attached to channel 15_8.

MAINTENANCE CHANNEL

The maintenance channel (MCH) is used for initialization of the CP and CM maintenance registers and monitoring of error status.

The maintenance channel consists of the maintenance channel interface on channel 17_8, a maintenance access control (MAC) in the CP, CM, IOU, and a set of interconnecting cables.

Any PP can be programmed to act as the maintenance control unit (MCU). However, hardware dictates PP0 as having special deadstart functions such that PP0 optimally serves as the MCU. In any case, the PP acting as the MCU performs initialization and maintenance functions that include:

- Initializing registers, controls, and memories
- Monitoring and recording error information
- Verifying error detection and correction hardware

The MCU directs these operations by sending function words (instructions) over the maintenance channel to the CP, CM, and IOU. The MCU retains all normal PP capabilities and, except for PP0 deadstart functions, does not gain any special hardware capabilities.

IOU MAINTENANCE REGISTERS

The MAC in the IOU contains several maintenance registers which hold IOU status or error information. Table I-2-8 lists the IOU maintenance registers. For detailed descriptions of these registers, refer to IOU Registers in volume II, section 2, listed in the System Publication Index in the preface.

Table I-2-8. IOU Maintenance Registers

Register Name	Number of Bits
Element Identification (EI)	32
Environment Control (EC)	32
Fault Status Mask	64
Fault Status 1	64
Fault Status 2	64
Options Installed (OI)	44
OS Bounds	64
Status Register	32
Test Mode	16

CENTRAL MEMORY ACCESS BY PPS

Any PP can access CM. During a write from the IOU to CM, the IOU assembles four successive 16-bit PP words into one 64-bit CM word (Virtual State) or five successive 12-bit PP words into one 60-bit CM word (CYBER 170 State). During a CM read, the IOU disassembles a 64-bit CM word into four 16-bit PP words (Virtual State), or a 60-bit CM word into five 12-bit PP words (CYBER 170 State).

To find the CM address, a PP reads the A register. If bit 17 of the A register is clear, the PP uses the contents of the A register for the CM address. If bit 17 of the A register is set, the PP adds the relocation address from the R register to the A register for the CM address.

A maximum of 20 PPs in various stages of assembly/disassembly can simultaneously read CM words, and 5 PPs can write CM words.

COMMENT SHEET

MANUAL TITLE: CYBER 170 Computer System Model 835
CYBER 180 Computer System Model 835
Virtual State Hardware Reference Manual Volume I

PUBLICATION NO.: 60469690 **REVISION:** A

NAME: _____

COMPANY: _____

STREET ADDRESS: _____

CITY: _____ **STATE:** _____ **ZIP CODE:** _____

This form is not intended to be used as an order blank. Control Data Corporation welcomes your evaluation of this manual. Please indicate any errors, suggested additions or deletions, or general comments below (please include page number references).

CUT ALONG LINE

FOLD ON DOTTED LINES

FOLD FOLD

CUT ALONG LINE

CONTROL DATA CORPORATION

Toronto Publications
1855 Minnesota Court
MISSISSAUGA, Ont., Canada
L5N 1K7

FOLD FOLD

CORPORATE HEADQUARTERS, P.O. BOX 0, MINNEAPOLIS, MINN. 55440
SALES OFFICES AND SERVICE CENTERS IN MAJOR CITIES THROUGHOUT THE WORLD

LITHO IN U.S.A.

CONTROL DATA CORPORATION